PRIJS ƒ 7.25/Bfr. 135

Commodore INFO

Commodore PC-1

VizaWrite

Beurs-Simulatie

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 4, NO. 6, OKT./NOV. 1987



LISTINGS

- Data-spin Amiga
- Shoot'em
- Yathzee
- Telex-writer
- Digi-ontwerper
- Libelle-Animatie
- Klok, schuinschrift
- Wiebel
- Piccolo C-16
- City-Destroyer

Computer Info Beurs 17 Oktober

Kinderen van de Wind

DTP op de Amiga

Hoge Resolutie 80-koloms

NOS Hobbyscoop

Vaste rubrieken
- Basic Miniatuurtjes
- Machinetaal
- Basic cursus
- Oud van Goudriaan
- Listings

Commodore-Info
Verschijnt 8 x per jaar

Jaargang 4, no. 6, okt/nov 1987

Uitgave:
Sala Communications

Uitgever:
Drs. J.Taverne

Redaktie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdredacteur

J. Boers, drs. M. de Rooij, drs. H. Zoete, P. Boncz, B. Munniksma, K. v.d. Vlies, drs. U. Schuurmans,

Redactiesecretariaat:
R. van Zalingen

Strip: Bert Tier
Illustraties: Ben van Mierlo, Ymmot

Advertentie-exploitatie:
Ing. V. Sala
Den Texstraat 5a
1017 XW Amsterdam
tel.: 020-273198

Redaktie adres:
Postbus 112
1269 AC Blaricum
tel.: 02152-65695

Abonnementen en administratie:
Postbus 5570
1007 AN Amsterdam
tel.: 020-273198
Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.

Abonnement:
Voor 8 nummers ƒ 47,50 of BFR. 950 per jaar. Betaling op Giro 1585491 t.n.v. SAC/COMMODORE-Info.
Oude nummers kunt U alleen krijgen bij vooruitbetaling van ƒ 6,75 op de bovenstaande rekening.
Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (Teleservice), elke dag tot 20.30 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor telefonische opgave van NIEUWE abonnementen.

Druk:
Verweij, Mijdrecht
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress/Gilze
In België: AMP/Brussel

ISSN: 0169-3085

Inhoud van dit nummer

Redactioneel
Commodore is altijd een bedrijf geweest, dat vele ijzers in het vuur had. Al in de begintijd kocht men een chipsfabrikant om de toevoer van chips voor de toenamlige PET en allerlei handrekenmachines veilig te stellen. Op een ander cruciaal moment, toen het succes van de Macintosh en de PC zich begon af te tekenen, verworf men het bedrijf Amiga Inc., omdat men daar zulke sublieme grafische en audio-chips had ontwikkeld voor een revolutionaire computer, die we nu allemaal als de Amiga kennen. Natuurlijk is er in de loop der jaren ook wel het een en ander mis gegaan, en bereikten bepaalde produkten niet de wereldtop.
Maar in het algemeen heeft Commodore getoond, over opmerkelijke vasthoudendheid te beschikken, waarmee men telkens weer uit het dal omhoog wist te klimmen. De situatie is nu, dat er in ieder geval weer een aardige winst gemaakt wordt en men met concurrerende produkten in vrijwel iedere (micro)-computercategorie aan de markt timmert. Met nog steeds een goed lopende Commodore 64 als absolute topper in aantallen, een nu weer met de PC-1 uitgebreide reeks MS-DOS modellen en als vlaggeschip de Amiga, heeft men een heel breed assortiment. Het is duidelijk, dat die machine in de loop der tijd toch een eigen publiek heeft gevonden, vaak gebruikers met bijzondere wensen die niet uit de voeten kunnen met een PC of Macintosh. Voor de komende maanden zien we daarbij een grote bloei voor de DeskTop Publishing op de Amiga, ook hier weer met net dat beetje extra (kleur), dat andere machines missen.

Luc Sala

Op de Seybold DTP-conferentie in Santa Clara was een Amiga systeem voor pagina-opmaak één van de noviteiten. De bezoekers van deze conferentie waren verbaasd over de grafische hoogstandjes en de snelheid, waarmee het beeld werd opgebouwd. De Amiga 2000 blijkt een krachtpatser op DTP gebied.

Desktop Publishing op de Amiga

Pagina-opmaak binnen bereik

Eens per jaar houdt Jonathan Seybold een grote conferentie over wat best de modegril van de tweede helft van de tachtiger jaren genoemd mag worden. Desktop Publishing ofwel met de computer zelf de drukwerkvoorbereiding, montage en (proef)-zetwerk doen is een grote doorbraak voor het computergebruik.

Het begon met Apple en de Apple Laserwriter, waardoor de computergebruikers, vooral in de VS, plotseling ontdekten, dat ze zelf professioneel uitziende publicaties konden maken. Het laatste jaar is de Apple wel ingehaald door de MS-DOS computers en zijn de meest bekende pakketten van de Apple ook voor de IBM-wereld beschikbaar gekomen. Op dit moment worden pakketten als Ventura in sneltreinvaart in bedrijven geïntroduceerd, omdat men wel inziet dat er grote besparingen en een duidelijke kwaliteitsverbetering van de presentatie mee bereikt kan worden.

'Total Presentation' systeem

De Amiga is van nature niet alleen geschikt voor het maken van fraaie pagina-layouts, maar kan nog veel meer met beelden doen. In Santa Clara was het thema rond de Amiga-stands dan ook vooral 'Total presentation', waarbij video, animaties, en DeskTop Publishing samen werden gebruikt om bepaalde boodschappen op een geïntegreerde manier veel effectiever te laten overkomen. Vooral die 'zelfuitgever' software staat in de belangstelling en in Santa Clara was dat ook het hoofdonderwerp.

Voor de Amiga zijn er een aantal voorzichtige pogingen geweest om DTP te lanceren en hier en daar zijn er ook wel pakketten getoond, die al heel wat konden bereiken. Met name Gold Disk Inc. met PageSetter was in dit opzicht de eerste met een werkbaar DTP-pakket. Maar van een DTP-doorbraak op de Amiga was nog geen sprake. Dat komt ook, omdat er een beetje een wanverhouding is tussen de prijs van een Amiga en die van een PostScript (de meest populaire vormtaal bij DTP programma's) compatibele laserprinter. De Amiga kost minder dan 2000 gulden, de printer meer dan 15 duizend. De gemiddelde Amiga-bezitter is dus niet in de markt voor zo'n printer en doet het met zijn matrix-model, met duidelijk mindere resultaten. Maar gelukkig komt de prijs van dergelijke laserprinters fors omlaag en gaat aan de andere kant de kwaliteit van de niet-PostScript laserpinters, zoals die van Canon, HP en Kyocera, ook omhoog.

Professional Page

Het nieuwste DTP pakket, en te zien als de opvolger van PageSetter, is ook van Gold Disk Inc. en heet Professional Page. Het is een compleet WYSIWYG pakket, in kleur en ook geschikt om kleurenseparaties voor vierkleurendrukwerk mee te maken, een unicum op DTP gebied.
Kwalitatief ook zeer goed is **Shakespeare** van Infinity Software, ook een compleet DTP pakket.
Microsearch heeft **City Desk** gemaakt, een pakket voor de Amiga, waarmee meerdere pagina's tekst tegelijk op het scherm gemanipuleerd kunnen worden, hetgeen voor complexe opmaakbezigheden als bij een krant natuurlijk van belang is.
Als laatste nieuwe DTP pakket noemen we Publisher 1000, van Brown-Wagh Publishing. Dat is een pagina-opmaak pakket voor nieuwsbrieven, brochures en dergelijke.



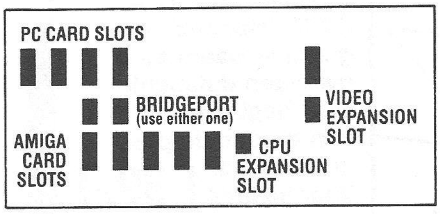
De Amiga 2000

TeX

De wetenschappelijke computerfans kennen zeker Knuth's tekstprogramma Tex, dat onder Unix een zeer wijdverbreid succes kent. Daarvan is er nu ook een Amiga versie, waarbij bijvoorbeeld ook typesetting-commando's in teksten kunnen worden 'ingebed'. Deze aanpassing van TeX is gemaakt door N-suqared Computer Consultants.

2000 met open architectuur

Natuurlijk is de Amiga 2000 de aangewezen machine voor het zwaardere werk, omdat door de vrije sleuven de koppeling van video-apparatuur via een gen-locker, MS-DOS uitbreiding en dergelijke het systeem heel universeel kunnen maken.



Zo werkt Bridgeport met PC en Amiga-slots

Om nu ook PC kaarten te kunnen gebruiken in een Amiga, is een aanpassing nodig, die onder de naam Bridgeport (A2088D) door het leven gaat. Hiermee is ook een zekere (kaart) hardwarecompatibiliteit met de PC bereikt. Kosten rond de 1.900 gulden.

Titeling

Om gemakkelijk ondertitels bij videoproducties, maar ook fraaie aftiteling met grafische effecten te maken, is tegenwoordig een Amiga 2000 met een kleurenmonitor en een A2300 genlock voldoende. Dat wil zeggen, het hiervoor meest geschikte programma, VideoTitler van Aegis Development Corporation, vraagt wel 2 MegaByte RAM aan geheugen.

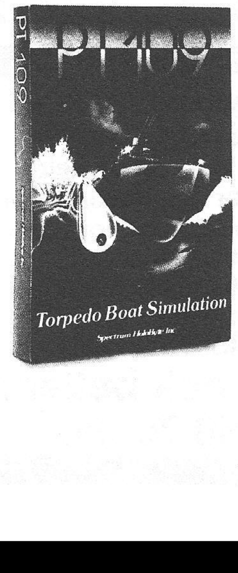
Networks

Een andere noviteit was het koppelen van Amiga's in een soort netwerk. Dergelijke 'peer-to-peer' koppelingen maken het gezamenlijk gebruiken van dure randapparatuur als een laserprinter veel gemakkelijker. Ook kan men 'files' gemakkelijk overbrengen van de ene computer naar de andere. De eerste LAN-kaart op dit gebied voor de Amiga is van Ameristar en laat koppeling aan een ARC-NET lokaal netwerk toe. Ook komen er netwerkprodukten beschikbaar om een koppeling naar een Ethernet tot stand te brengen, die werkt volgens het TCP/IP en UDP/IP protocol. De koppeling naar de grotere computers is daarmee mogelijk.

SOFTWARE

Torpedo Boat Simulation

In de Verenigde Staten is zojuist het pakket Torpedo Boat Simulation verschenen. Dit speciaal voor de C-64 gemaakte wargame speelt zich geheel op het water af. Wie van snelheid en actie houdt en niet bang is om nat te worden, wordt bij deze uitgenodigd om aan boord van de PT-109 patrouilleboot te komen. Door zijn uitstekende manouvreerbaarheid kan de PT boot in verscholen baaien en andere gebieden komen waar andere schepen geen toegang hebben. Maar zelfs daar ben je niet veilig, de vijand is niet alleen op het water aanwezig, maar schuilt ook in de lucht, dus pas goed op, je hebt al je wapens nodig om te kunnen overleven. In het ene geval zullen de torpedo's beter van pas komen dan de dieptebommen, maar ook de raketten en de machinegeweren zullen hun diensten bewijzen. Overleven is in deze wa(te)rgame een moeilijke opgave.



Torpedo Boat Simulation heeft vier moeilijkheidsgraden en verder zijn er vele verschillende opdrachten, die door de opperbevelhebber kunnen worden opgedragen. Jammer voor onze Hollandse mijnvegers die momenteel in de Perzische golf ronddobberen, maar het pakket is momenteel nog niet in Nederland verkrijgbaar; het zal echter niet lang meer op zich laten wachten. In de VS is het momenteel verkrijgbaar voor $35, wat de prijs hier zal worden is nog niet bekend.

The Armageddon Man

Armageddon Man van Martech zal waarschijnlijk in augustus op de markt komen met een Commodore versie. Deze game is echt geen simpele shoot'em up, maar meer een combinatie van computeractie met tactiek en mentale kracht.

Armageddon Man speelt zich af in het jaar 2032. Het nucleair wapenarsenaal heeft zich de laatste dertig jaren ernstig uitgebreid. Er zijn inmiddels 16 nucleaire supermachten, elk met het vermogen om een beging te maken met het einde van de wereld (Armageddon).

De Armageddon man is de opperbevelhebber van de United Nuclear Nations, en hij heeft een enorm netwerk ter beschikking om alle gebeurtenissen op politiek, sociaal en militair niveau te controleren. De Armageddon man heeft tot taak om een nucleaire oorlog te voorkomen, maar mag in het ergste geval gebruik maken van zijn laserwapens om gevaarlijke conflicten in de kiem te smoren. Voor de liefhebber van Risk of mensen met grootheidswaanzin een ideaal spel. Het spel, van Martec, zal ongeveer 35 gulden gaan kosten.



Armageddon man

Na het artikel in Commodore Info 4/87 over het maken van een multiple choice-test op de computer nu deel zes van de reeks over leerprogramma's. Het onderwerp is deze keer een simulatie-spel.

Zelf leerprogramma's maken in BASIC

Deel 6: Beurs-simulatie

Er zijn in deze reeks al heel wat verschillende doelgebieden voor computergebruik binnen het onderwijs aan de orde geweest. Deze keer toetsen we de computer op een ander toepassingsgebied. We richten onze aandacht deze maand op het nabootsen van de werkelijkheid: een simulatie van het beursgebeuren.

Zoals al eerder bleek zijn er talloze mogelijkheden om een computer zinvol binnen het onderwijs in te zetten. Zelfs een avontuur in de klas bleek voor veel lezers een spannende uitdaging. Bedankt voor alle reacties en opmerkingen hierover. Wij zijn blij met uw betrokkenheid. Deze keer gaan we eens in op een ander gebied van het computergebruik in de klas, namelijk het simuleren van mogelijk werkelijke situaties.

Een voorbeeld

In andere vormen van onderwijs, vooral in het wetenschappelijk gericht onderwijs, is het simuleren van een bepaalde situatie, voor studiedoeleinden, vrij gebruikelijk. Vooral als de bewuste situatie erg gevaarlijk of kostbaar is. Een computer is dan een snelle, goedkope en vooral veilige manier om de effecten en resultaten van bepaalde beslissingen te kunnen bestuderen. Op universiteiten staan reeksen computers alleen voor dit doel opgesteld. Het nagaan van bijvoorbeeld genetische veranderingen onder invloed van radioactieve straling laat zich goed in de praktijk onderzoeken. Men neemt daarvoor vaak fruitvliegjes, die een korte levenscyclus hebben en zich snel vermenigvuldigen (1 dag). Eventuele mutaties zijn zeer snel zichtbaar. Met behulp van een computer kan men op grond van de uitkomsten de veranderingen in de komende honderden generaties van dat vliegje met zeer grote nauwkeurigheid voorspellen. En hoewel dit voorbeeld natuurlijk niet spectaculair te noemen is, is het wel maatgevend voor de ongekende gebruiksmogelijkheden van de computer op dit gebied. Door middel van een programma, dat een situatie simuleert op grond van bekende gegevens, kan men inzicht verkrijgen in bepaalde processen, die bepalend zijn voor het eindresultaat. Daarbij heeft een simulatie de voordelen van veiligheid, betaalbaarheid en snelheid. Comfortabeler is het ook nog in de meeste gevallen.



Simulatie in de klas

Wat moeten we nu met een simulatie in de klas. Dat is een duidelijke vraag. Het antwoord wordt net zo duidelijk, als we proberen de vakgebieden, waarin we als onderwijsgevende zelf vaak proberen aanschouwelijk onderwijs te geven, te omschrijven. Vaak zijn er abstracte zaken aan de orde, die de aanschouwelijkheid behoeven, omdat alleen dan de situatie voor de kinderen helder wordt. Dat is zeker in de hogere klassen van de basisschool zo. Vaak gebruiken we dan een programma op videoband of audiocassette of we presenteren een dia-serie. Bij projecten e.d. komt er nog wel eens een gast-docent. Sommige situaties krijgen helderheid door het spelen van een rollenspel of de groep gaat op excursie. En dan tenslotte, bij sommige zaken kan de computer een handje helpen. Al deze genoemde dingen zijn normale routine in het klassegebeuren. Computergebruik in deze helaas nog niet. Commodore-Info wil een bijdrage leveren, om hierin verandering te brengen.
Er zijn veel zaken die geschikt zijn om gesimuleerd te worden. Ga zelf de mogelijkheden maar na. Van vliegsimulatie, maanlander tot meer laag-bij-de-grondse aktiviteiten zoals bijvoorbeeld het beveiligen van een gebouw of het regeren over een denkbeeldig

land. Of meer technische zaken zoals het simuleren van de werking van een atoomcentrale of een verbrandingsmotor, overal kan de computer van grote dienst zijn. Wij kozen voor een beurs-simulatie. Aan de hand van een door ons ontwikkeld beurs-simulatie programma, willen we de mogelijkheden van simulatie in de schoolpraktijk in een ander daglicht plaatsen.

De effectenbeurs

Op een effectenbeurs verhandelt men aandelen en andere waardepapieren. De prijs wordt grotendeels bepaald door de 'stemming' en de verhouding tussen vraag en aanbod. Deze mechanismen zijn redelijk te overzien, dus ook te simuleren. Verder kunnen allerlei onverwachte schommelingen in de koersen worden veroorzaakt door meer willekeurige factoren. Verder kunnen fondsen zo slecht lopen, dat de prijs tot een minimum daalt of het gaat zo goed, dat er dividend wordt uitgekeerd aan de bezitters van de bewuste aandelen. Uitgaande van een nominale koers van 100 kan de waarde van een aandeel, afhankelijk van de koers, dalen of stijgen. De koers is dan het waarde-percentage. Bekende materie, wellicht.

Zie daar alle ingredienten voor een simulatie van de beurs voor kinderen in de klas, die door middel van de simulatie enig inzicht krijgen in het reilen en zeilen van een effectenbeurs. Het wordt gespeeld als educatief spel. Let wel, spel, want het gaat slechts om een gesimuleerde situatie.

Het programma

Het programma is zo gemaakt, dat het scherm steeds dezelfde indeling heeft. Dat geeft rust en alles is direct te overzien. Het onderste deel van het scherm wordt gebruikt voor wisselende mededelingen. Steeds als er een handeling wordt verricht, loopt het programma door een koersbepalende module. Die module staat verder borg voor de verrassingen voor de speler. Heel eenvoudig kunt u dus zelf het verloop beïnvloeden, door in die programma-module veranderingen aan te brengen. Maar laten we niet op de zaken vooruit lopen. Ik wil met u het programma stap voor stap doornemen, zodat de programmeer-strategie duidelijk wordt en u de opgedane ervaringen ook in andere simulaties kunt benutten. Want daar gaat het ons wel om: het doorgeven van ervaringen ter meerdere glorie van de computer in de klas.

Vaste technieken

We gebruiken een aantal vaste programmeer-technieken, die u wel zult herkennen uit de voorgaande afleveringen. We maken eerst een aantal strings klaar om te gebruiken bij de cursor-plaatsing: NE$ en RE$. Ook komt de spatie-string SP$ erg van pas. Zie daarvoor de regels 155-170. Omdat ook de functie-toetsen steeds weer een functie hebben in het programma, evenals de RETURN-toets, zetten we de aanduidingen ook in een aantal strings. Zie regel 195-215.

Regel 230-255 realiseert het wachten op een toets-indruk en kan als subroutine in elk deel van het programma worden aangeroepen. Klok op nul en daar gaat het programma van start. De eigenlijke 'body' van het programma staat in de regels 265-285. U ziet wel, dat het programma voornamelijk bestaat uit diverse sub-routines, die in een bepaalde volgorde worden uitgevoerd. Er zitten grote voordelen aan deze manier van modulair programmeren. Heel eenvoudig kan de loop van een programma worden bepaald en gewijzigd, als die niet bevredigend funktioneert. Voor alle handelingen zijn er sub-routines, die u kunt beschouwen als afgeronde mini-programmaatjes, die ook zelfstandig kunnen draaien. Dat is vooral handig bij het uittesten van het programma. Elke sub-routine zou los kunnen worden ingetikt, getest en bij goedbevinden worden weggeschreven. Later kunnen de modules dan via een APPEND-commando (u heeft dan wel een goede BASIC-toolkit nodig!) aan elkaar worden geplakt. Maar goed, dat hoeft natuurlijk niet.

OPROEP

Bij deze aflevering, waarin wij u op zinvolle ideëen willen brengen voor het zelf ontwikkelen van programmatuur voor in de klas, een verzoek. Wij zijn er van overtuigd, ook gezien de reakties van de lezers, dat er veel zelfgemaakte schoolprogramma's op diverse tapes en diskettes staan. Ieder die zich met de materie bezighoudt heeft wel zo zijn eigen inzichten en bekwaamheden. Commodore-Info wil als bemiddelaar optreden, om deze programma's onder een breder publiek te brengen. Waarom zwoegen aan een bepaald programma, als een ander dat allang heeft uitgewerkt. Als iedereen, die zelf leerprogramma's ontwikkelt (hoe bescheiden ook), zijn of haar produkten naar onze redaktie stuurt, dan verzamelen wij alle ingestuurde programma's en bieden ze dan op basis van kostprijs als totaalpakket aan onze lezers aan. Zo kan een omvangrijke programmabank ontstaan, waar iedereen zijn of haar voordeel mee kan doen. Doe allemaal mee, dan kan het een groot succes worden. Stuur uw leerprogramma's op cassette of diskette naar:

Redaktie Commodore-Info
Postbus 112
1260 AC Blaricum

onder vermelding van **"Leerprogramma's".**

Schermpresentatie

Zo wordt de schermopbouw geregeld in de module die begint op regel 295. Hoewel we in eerdere programma's uitsluitend gebruik maakten van CHR$-codes bij het opzetten van kaders en dergelijk, stappen we in dit programma even van die conventie af. De reden is puur praktisch. Een eenvoudig kader laat zich met CHR$-codes gemakkelijk opbouwen. Het kader in dit simulatie-programma is wat te ingewikkeld voor deze werkwijze. U zult nu gebruik moeten maken van de tekenset van uw Commodore 64. Voor op de lettertoetsen staan allerlei grafische tekens. U kunt die op het scherm zetten, door op de SHIFT- of Commodore-toets te drukken, tegelijk met die bewuste toets. Raadpleeg even de gebruikers-handleiding van uw computer. Denk er wel aan, dat de computer in onderkast-letterstand moet staan. Dat scheelt wel in het aantal tekens. Het scherm is 38 karakters breed. Er blijven dus 2 posities over aan de rechterkant. Dat bleek eenvoudiger bij de schermpresentatie. Regel 390 wist acht keer een regel onder het eerder geplaatste kader.

De sub-routines

Zoals gezegd bestaat het programma uit een aantal sub-routines. De schermopbouw en het wachten op een toets-indruk (resp. regel 295 en 230) kwamen al ter sprake. Maar er zijn er natuurlijk meer. Op regel 405 start het invullen van de eerder geplaatste scherm lay-out: de financiële status van de speler tot in detail. Daarbij wordt een schijnbaar merkwaardige konstruktie gebruikt om de gegevens op het scherm te printen. Regel 430 lijkt wel een toverformule uit het land van Oz. Niets is minder waar. Omdat in de BASIC van de C-64 een PRINT USING instructie ontbreekt om

getallen mooi te rangschikken, doen we het op deze manier. We gebruiken dezelfde formule verschillende malen. Lees even mee:

430 print left$(re$,5-len(str$(kb(i))));
440 print kb(i);

De string RE$ werd gebruikt voor het plaatsen van de cursor op een bepaalde kolom op het scherm. Zoiets doet TAB(x) ook. Nu halen we een bepaald deel uit de RE$-string om de eerstvolgende print-positie vast te stellen. Die positie is vervolgens afhankelijk van de lengte van STR$(kb(i)), die wordt afgetrokken van de vaste waarde 5. Ter verduidelijking een voorbeeld met echte getallen. Stel kb(i), dat is het aandelenbezit van een bepaald fonds, is 12. De verwerking van regel 430 is dan zo:

a. **430 print left$(re$,5-len (str$(kb(i))));**
b. **430 print left$(re$,5-len(" 12"))**
c. **430 print left$(re$,5-3)**
d. **430 print left$(re$,2)**

Na de bewerking met STR$(x) is die bewuste string drie karakters lang. Let op de spatie die er door de bewerking bijkomt! De print-positie wordt zo twee spaties naar rechts bepaald. Schrik niet. U hoeft dit niet zo in te typen. De oorspronkelijke regel 430 voert alle handelingen automatisch uit. Het effect is natuurlijk duidelijk. Alle eenheden, tientallen en honderdtallen van de waarden op het scherm komen keurig recht onder elkaar. Dat ziet er dan netjes en overzichtelijk uit.
Er zijn in de simulatie vier hoofdverrichtingen. Dat zijn kopen, verkopen, het bekijken van het aandelenaanbod en het laten passeren van een beurt (niets doen dus). Vanuit het hoofdmenu kan de speler kiezen. Zie daarvoor de sub-routine die begint op regel 530. De funktie-toetsen doen het werk in de sub-routine vanaf regel 585. Afhankelijk van de ingedrukte funktie-toets verspringt het programma naar de betreffende onderdelen.

Kopen

Het kopen van aandelen begint op regel 830. De beschikbare fondsen verschijnen op het onderste gedeelte van het scherm. De speler moet het gewenste nummer indrukken. Zijn er aandelen van dat fonds te koop (m.a.w. is er inderdaad aanbod), dan kan de transactie verder worden afgehandeld. Is dat niet zo, dan verschijnt de overeenkomstige mededeling en de speler moet opnieuw een keuze uit het hoofdmenu doen. Hij of zij zou natuurlijk eerst even het aanbod kunnen bekijken via funktie-toets f7.
Als er aanbod is, moet de speler het aantal te kopen aandelen bepalen. Dit kan eenvoudig door te drukken op de '+'-toets voor verhogen of op de '-'-toets voor verminderen van het aantal. Is het gewenste aantal bereikt, dan wordt na een druk op RETURN gekeken of de speler wel voldoende geld heeft voor de aankoop. Zijn de financiën in orde, dan wordt de transactie definitief uitgevoerd, de koerswijzigingen berekend en tenslotte wordt het schermkader voorzien van de nieuwe gegevens.

Verkopen

Het verkopen heeft ook een vast programma-verloop. Eerst wordt gevraagd van welk fonds verkocht moet worden. Indien die in bezit zijn, moet de speler aangeven hoeveel er verkocht moeten worden. Als alle gewenste handelingen toegestaan zijn, wordt de transactie verder afgehandeld, de koers aangepast en het scherm opnieuw ingevuld. Ook nu meldt het programma een niet mogelijke transactie en vertelt ook de reden waarom niet.

Geen transactie en aanbod

Ook kan het soms slim zijn een bepaalde koersontwikkeling af te wachten zonder iets te kopen of te verkopen. Dat kan ook in deze simulatie. Let op, het kan soms ook heel onverstandig zijn om niets te doen!
Kiest men met deze optie uit het hoofdmenu, dan krijgt men een overzicht van de aangeboden aandelen, de aantallen en hun koers. Verder heeft deze keuze geen invloed op het koersverloop.

De overige modules

Verder heeft het programma een aantal sub-routines, die het verloop van het programma ondersteunen en dus van wezenlijk belang zijn. Vanaf regel 775 wordt een lopende klok op het scherm gezet en bepaalde voorwaarden worden getest. Bij een bepaald maximum of minimum bedrag aan geld of een bepaalde kloktijd wordt het spel gestaakt met daarbij behorende mededelingen. Die spreken verder voor zich.
Als de koers van een aandeel boven de 300 nominaal komt, dan ontvangt de speler het boventallige bedrag als dividend op zijn of haar bankrekening. De koers wordt begrensd tot 300. Dat valt te lezen in de sub-routine vanaf regel 740. Daalt de koers beneden 1, dan bent u uw aandelen kwijt en de koers wordt op de startwaarde 10 gezet. Uitkijken geblazen dus.
Vanaf regel 675 worden de koersfactoren verwerkt. Die module is natuurlijk bepalend voor het hele spel. We hebben de berekeningen niet al te ingewikkeld gemaakt. Ten eerste om het geheel begrijpelijk te houden. Ten tweede omdat wij ervan overtuigd zijn, dat er lezers zijn, die dat veel beter kunnen.
Aan het einde van het programma vindt u de startwaarden en andere vaste gegevens van het programma. Niets houdt u tegen om daarin naar eigen inzicht verandering te brengen. Ook andere begrenzers (bijvoorbeeld dividend, tijd, max. en min. geldbedrag) kunt u zelf anders bepalen.
Tot zover deze aflevering. De volgende keer komt de Abacus aan de orde.

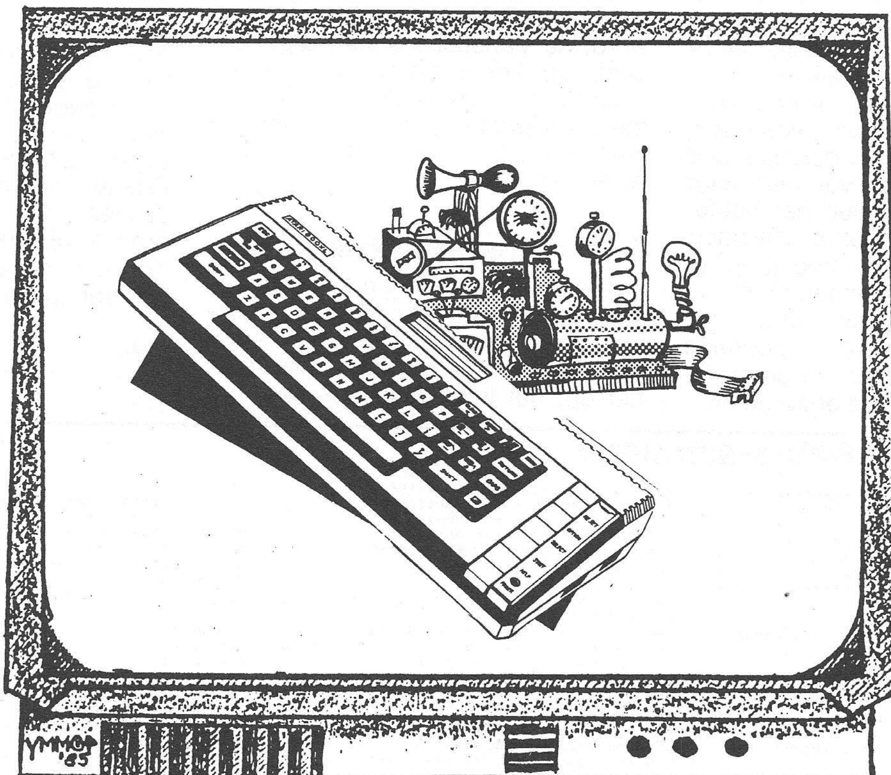
B.M. •

Listing Beurs-simulatie

```
100 rem ******************************
105 rem **   beurs simulatie spel    **
110 rem **                           **
115 rem **  leerpramma's in basic    **
120 rem ** (c) 1987  bob munniksma   **
125 rem **   voor commodore info     **
130 rem ******************************
135 :
140 rem initialisatie
145 poke 53280,0:poke 53281,0:poke646,7
150 print chr$(14)chr$(8)chr$(147);
155 ne$=chr$(19)
160 for i=1 to 30:ne$=ne$+chr$(17)
165 re$=re$+chr$(29)
170 sp$=sp$+chr$(32):next i
175 for i=1 to 4
180 read kr$(i),kr(i),ab(i),kb(i)
185 next i:read fl
190 for i=1 to 6:read kn$(i):next i
195 f1$=chr$(18)+"f1"+chr$(146)
200 f3$=chr$(18)+"f3"+chr$(146)
205 f5$=chr$(18)+"f5"+chr$(146)
210 f7$=chr$(18)+"f7"+chr$(146)
215 r$=chr$(18)+" RETURN "+chr$(146)
220 ti$="000000":goto 265
225 :
230 rem wachten op een toetsindruk
235 gosub 775
240 print left$(ne$,23)left$(re$,1);
245 print" Druk op "r$" om verder te gaan"
250 get t$:if t$="" then 250
255 return
260 :
265 rem stuurgedeelte
270 gosub 295:gosub 405
275 gosub 530:gosub 585
280 on t gosub 830,960,1040,1165
285 gosub 675:s=s+1:goto 265
290 :
295 rem beeldscherm opbouw
300 print chr$(19);
305 print"┌───────────────────────────────────┐"
310 print"│     Beurs     Simulatie    Spel   │"
315 print"├────────────┬──────┬──────┬────────┤"
320 print"│ Fonds      │Bezit │Koers │ Waarde │"
325 print"├────────────┼──────┼──────┼────────┤"
330 print"│            │      │      │        │"
335 print"│            │      │      │        │"
340 print"│            │      │      │        │"
345 print"│            │      │      │        │"
350 print"├────────────┴──────┴──────┴────────┤"
355 print"│ Geld in Kas  : fl.                 │"
360 print"│ Waarde bezit: fl.                  │"
365 print"│ Stappen      :                     │"
370 print"│ Tijd         :                     │"
375 print"└───────────────────────────────────┘"
380 print left$(ne$,16);
```

Vervolg listing beurs-simulatie

```
385 for i=1 to 8
390 print"
395 next i:return
400 :
405 rem financiele status
410 tw=0
415 print left$(ne$,5)
420 for i=1 to 4:if kb(i)=0 then 470
425 print left$(re$,2)kr$(i);
430 print left$(re$,6-len(str$(kb(i))));
435 print kb(i);
440 print left$(re$,5-len(str$(kr(i))));
445 print kr(i);
450 w(i)=kb(i)*kr(i)
455 print left$(re$,9-len(str$(w(i))));
460 print w(i)chr$(157)",-"
465 tw=tw+w(i)
470 next i
480 print left$(ne$,11);
485 print left$(re$,32-len(str$(fl)));
490 print fl;chr$(157)",-"
495 print left$(ne$,12);
500 print left$(re$,32-len(str$(tw)));
505 print tw;chr$(157)",-"
510 print left$(ne$,13);
515 print left$(re$,32-len(str$(s)))s
520 return
525 :
530 rem menu
535 print left$(ne$,15)
540 print left$(re$,3)"Maak een keuze:"
545 print left$(ne$,19)left$(re$,3);
550 print f1$;"  Kopen"tab(20);
555 print f3$;"  Verkopen"
560 print left$(ne$,21)left$(re$,3);
565 print f5$;"  Geen aktie"tab(20);
570 print f7$;"  Aanbod"
575 return
580 :
585 rem opdracht ophalen
590 gosub 775
595 get t$:if t$="" then 590
600 t=asc(t$)-132
605 if t<1 or t>4 then 590
610 return
615 :
620 rem geld genoeg
625 ks=a*kr(f)
630 if fl<ks then kn=1:gosub 910:return
635 fl=fl-ks:kb(f)=kb(f)+a
640 ab(f)=ab(f)-a:return
645 :
650 rem verkoop uitrekenen
655 ab(f)=ab(f)+a:os=a*kr(f)
660 if h=kb(i) then ct=int((a*kr(i)/100)
665 fl=fl+os-ct:kb(f)=kb(f)-a:return
670 :
675 rem koersfactoren verwerken
680 if t=1 then ab(f)=ab(f)-a
685 if t=2 then ab(f)=ab(f)+a
690 gosub 380:for i=1 to 4
695 if ab(i)<1 then ab(i)=1
700 if ab(i)>20 then ab(i)=20
705 kr(i)=int(kr(i)*(10/ab(i)))
710 if kr(i) < 1 then kr(i)=10:kb(i)=0
715 if kr(i) > 300 then gosub 740
720 if t=3 then ab(i)=ab(i)+10:goto 730
725 ab(i)=ab(i)+int(rnd(1)*20)-10
730 next i:return
735 :
740 rem dividend
745 dv=kr(i)-300:kr(i)=300
750 fl=fl+kb(i)*dv
755 x=int(rnd(1)*50)
760 if t=3 then kr(i)=kr(i)-x
765 return
770 :
775 rem tijd ophalen en controleren
780 print chr$(18);
785 print left$(ne$,14)left$(re$,25);
790 print mid$(ti$,2,1)":";
795 print mid$(ti$,3,2)":"right$(ti$,2)
800 rem controleren
805 if fl>999999 then kn=5:goto 910
810 if fl<1 and tw<1 then kn=4:goto 910
815 if val(ti$)>1000 then kn=6:goto 910
820 return
825 :
830 rem kopen
835 gosub 380:print left$(ne$,16);
840 print left$(re$,3)"Kies een fonds:"
845 print left$(ne$,18);:for i=1 to 4
850 print left$(re$,4);i;kr$(i):next i
855 gosub 250:f=val(t$)
860 if f<1 or f>4 then 855
865 h=ab(f):gosub 380
870 if h<1 then kn=2:goto 920
875 print left$(ne$,16)left$(re$,3);
880 print chr$(18)" Kopen "chr$(146);
885 print left$(re$,4)kr$(f)
890 print left$(ne$,18)left$(re$,5);
895 print"Aantal: "
900 gosub 1055:gosub 620:return
905 :
910 rem geen transactie mogelijk
915 gosub 380
920 print left$(ne$,16)left$(re$,2);
925 print chr$(18)" Geen transactie "
930 print left$(ne$,20);
935 print left$(re$,2);kn$(kn)
940 gosub 240
945 if kn=4 or kn=5 or kn=6 then run
950 return
955 :
960 rem verkopen
965 gosub 380:print left$(ne$,16);
970 print left$(re$,3)"Kies een fonds:"
975 print left$(ne$,18);:for i=1 to 4
980 print left$(re$,4);i;kr$(i):next i
985 gosub 250:f=val(t$)
990 if f<1 or f>4 then 985
995 h=kb(f):gosub 380
1000 if h=0 then kn=3:goto 920
1005 print left$(ne$,16)left$(re$,3);
1010 print chr$(18)" Verkopen "chr$(146);
1015 print left$(re$,4)kr$(f)
1020 print left$(ne$,18)left$(re$,5);
1025 print"Aantal: "
1030 gosub 1055:gosub 650:return
1035 :
1040 rem geen transactie
1045 return
1050 :
1055 rem aantal bepalen
1060 a=1
1065 print left$(ne$,18);
1070 print left$(re$,16);
1075 print left$(sp$,5-len(str$(a)));a
1080 print left$(ne$,20)left$(re$,2);
1085 print chr$(18)"+"chr$(146)" = Meer";
1090 print left$(re$,2);
1095 print chr$(18)"-"chr$(146)" = Minder";
1100 print left$(ne$,22)left$(re$,2);
1105 print r$" = Transactie"
1110 gosub 250
1115 if asc(t$)=43 then a=a+1:gosub 1140
1120 if asc(t$)=45 then a=a-1:gosub 1140
1125 if asc(t$)=13 then return
1130 goto 1065
1135 :
1140 rem controle
1145 if a < 1 then a=1
1150 if a > h then a=h
1155 return
1160 :
1165 rem aanbod
1170 gosub 380:print left$(ne$,16);
1175 print left$(re$,2)chr$(18);
1180 print" Aanbod en koers "
1185 print left$(ne$,17):for i= 1 to 4
1190 if ab(i)<=0 then 1220
1195 print left$(re$,5)kr$(i);
1200 print left$(re$,6-len(str$(ab(i))));
1205 print ab(i);
1210 print left$(re$,6-len(str$(kr(i))));
1215 print kr(i)
1220 next i:gosub 230:gosub 380:return
1225 :
1230 rem gegevens in dataregels
1235 rem startwaarden
1240 data"Spoorwegen",100,10,0
1245 data"Bankwezen ",100,07,0
1250 data"Luchtvaart",100,02,0
1255 data"Computers ",100,04,0
1260 data 300
1265 data"Je hebt onvoldoende geld in kas"
1270 data"Er is geen aanbod van fonds"
1275 data"Je hebt geen aandelen van dit fonds"
1280 data"Het spel is uit! Je bent failliet"
1285 data"Het spel is uit! Je bent miljonair"
1290 data"Het spel is uit! De tijd is om"
```



Op de Efficiency vakbeurs in de RAI was vooral duidelijk, dat de PC en AT's op zakelijk gebied vrijwel door iedereen als standaard geaccepteerd zijn. In sommige deelgebied, zoals DTP met de Macintosh en DeskTop Video met de Amiga, houden de afwijkende machines nog stand, maar ook daar is de mS-DOS emulatie een verplicht nummer. Commodore toonde niet de PC-1, maar was wel aanwezig met een brede reeks verticale software- oplossingen.

Efficiency Vakbeurs '87

Matig succes en weinig nieuws

De EVB trok dit jaar, na een zeer matige start op de eerste twee dagen, volgens de RAI/Vifka toch nog bijna 80.000 bezoekers. Maar gezien de steeds maar stijgende dagbezoekgrafiek leek het er vooral de laatste dag op, dat men de zaak ook wel erg graag op die vooraf aangkondigde 80.000 bezoekers wilde brengen. Het feitelijke monopolie van de brancheverenigingVifka/RAI op dit computerbeurzengebied is echter sterk, mogelijke kritiek durven de standhouders niet erg duidelijk te melden. In de wandelgangen kon men wel horen, dat erg veel bedrijven helemaal niet zo blij zijn met elk jaar een Efficiency (Vak)beurs, maar het voorbeeld van IBM, HP en Wang niet durfden te volgen (die bleven gewoon weg van de beurs).

Afgeblazen Micro-Trofee

Op de eerste avond stond de uitreiking van de Micro-Trofee van de VNU op het programma. Men heeft dat echter op het laatste moment afgeblazen, om vrij onduidelijke redenen, maar daarmee leed deze uitgever een gevoelig gezichtsverlies bij de deelnemers aan de EVB. Bij geruchte zou deze prijs naar de Adventure-clan van Radarsoft gaan, die natuurlijk bij ons collega-blad op Commodore gebied ook wel erg goed vertegenwoordigd is. Maar dit jaar dus geen Micro-Trofee. Een op zich aardige traditie is in schoonheid gestorven.



De Commodore-stand van dit jaar

Beperking

De ongeveer 400 deelnemers aan de Efficiency Vakbeurs werden dit jaar gedwongen zich te beperken tot een oppervlakte van 250 vierkante meter per stand. Dat gegeven zorgde, in tegenstelling tot wat verwacht werd, voor een minstens net zo luxueuze aanpak als in de afgelopen jaren. De 'grote' stands zagen er meestal prachtig uit.

PTT in de aanval

De privatisering van de PTT heeft nu geleid tot een offensief op het gebied van de telecommunicatie. Onder het motto *'telegration'* (een onuitsprekelijk woord; speciaal voor de export?) presenteerde de PTT een concept waarbij telecommunicatie en automatisering hand in hand moeten gaan. Daarbij speelt Memocom, de electronische berichtendienst van de PTT, een belangrijke rol. Ondertussen is wel duidelijk, dat de ooit als revolutionair gebrachte Viditel-dienst het niet gemaakt heeft. Men wil nu afstappen van de pagina-gewijs berichten en op de meer universele ASCII tekstmode gaan werken. Daarmee wordt Viditel dan net zo'n soort dienst als "The Source" in de VS en de fraaie plaatjes, die met Viditel wel, maar met ASCII niet kunnen worden overgedragen, verdwijnen dan. De PTT loopt hier natuurlijk niet erg mee te koop, maar men is al een jaar bezig de oude voorraad Viditel-modems tegen steeds lagere prijzen op te ruimen. Dat gaat dan om de V21-V23 modellen, die dus met de ondertussen befaamde 1200/75 baud snelheid werken. Voor meer serieus dataverkeer, met bijvoorbeeld electronic mail-verkeer in twee richtingen, stapt men nu meestal over op V22, met 1200/1200 baud kan men dan in beide richtingen net zo snel ver-

sturen. Daarmee is de V21/V23 techniek wat achterhaald, al is het jammer, dat met name voor de Commodore 64 dit nog wel het meest gebruikte modem is. Als pleister op de wonde kan men dan ook nog wel met 300/300 baud gaan werken (de V21 mode), maar dat is vrij langzaam. De PTT heeft hier eigenlijk een wat laakbare opstelling gekozen, is van zijn oude rommel af, maar daarmee is de gebruiker niet echt geholpen. Voor een bedrijf, dat over ruim een jaar als zelfstandig en commercieel opererend bedrijf de markt op moet, een kortzichtig beleid.



PC 40/40

Commodore was op de Efficiency Vakbeurs vooral vertegenwoordigd met zakelijke toepassingen op de Commodore computers, vooral de PC-lijn, waarbij de PC 40/40 als nieuw model werd geïntroduceerd. Op de Amiga werden twee nieuwe professionele pakketten getoond, die nog in een beta-versie draaiden. Professional Page is de opvolger van PageSetter, en veel viel er nog niet over te zeggen, omdat het pakket pas enkele dagen in Nederland was. Een student van de kunstacademie Utrecht liet wat van de mogelijkheden zien, zoals het inlezen van de files van het eveneens geïntroduceerde Wordperfect (4.1) voor de Amiga.

CAD op Amiga

Iets beter te overzien waren de mogelijkheden van het **XCAD**-pakket, een qua capaciteiten met AutoCAD (voor MS-DOS) vergelijkbaar CAD-pakket. Met een prijs van ongeveer 2.000 gulden echter veel goedkoper dan AutoCAD, waarmee Commodore dus een leuke troef in handen heeft in de strijd om de zakelijke computerkoper. Ook XCAD werd nog gedemonstreerd in een demoversie, waarmee nog niets kon worden bewaard, terwijl er enkele keren een 'software-failure' optrad.

Binnen enkele weken is het volgens distributeur KPD wel leverbaar. Het systeem zal diverse printers en plotters ondersteunen.

De PC1 was slechts achter de gordijnen te bewonderen, en we verwachten dat model voor het eerst op de Computer Info Beurs op de RAI op 17 oktober zal worden getoond.

Commodore is, nadat we het bedrijf misten op de PC-Expo in New York begin september, in Europa in ieder geval duidelijk actief aanwezig in de markt. In Parijs toonde commodore zich met een behoorlijk grote presentatie op de mini-SICOB, waar we de primeur van de PC 1 weghaalden.

De Najaarsbeurzen

Commodore met PC-1 op Parijse computerbeurs

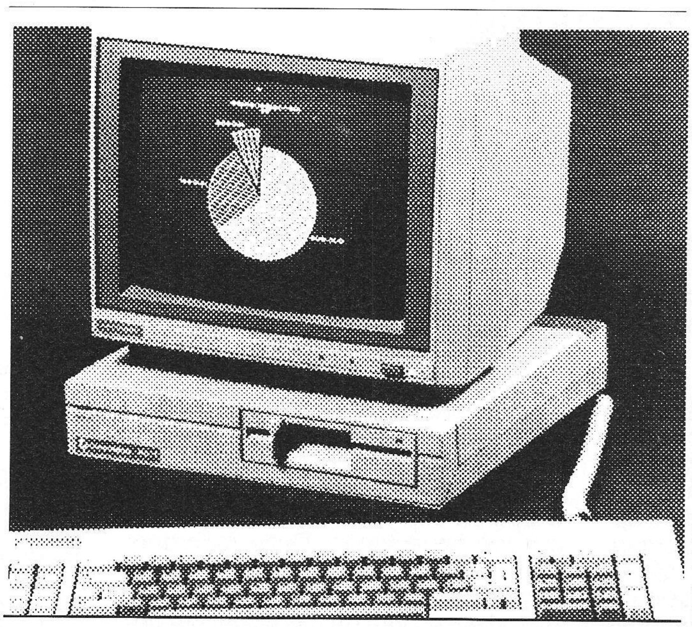
De 'Grande Exposition de la Micro Informatique', die van 14 tot 19 september werd gehouden in de Parijse buitenwijk La Defense, werd beheersd door de Atari stand. Deze stand, waarin een aantal software-leveranciers een plaatsje hadden gevonden, nam ruim een vijfde van het totale beursoppervlak in beslag. De grote namen als IBM, Hewlet Packard, Apple en dergelijke ontbraken in het rijtje van 187 exposanten, waarvan behalve Atari ook Commodore een grote stand innam. I

In Frankrijk lijken deze twee giganten van de huiscomputers ook op de zakelijke markt te gaan doordringen, met respectievelijk de Mega-ST en de Amiga 2000. Er waren diverse Franse softwarepakketten voor deze machines te zien, waarmee deze apparaten in de kleine (en zakelijke) markt gedegen concurrenten lijken te gaan worden voor de PC's en de Macintosh.

Atari laserprinter

Het verrassende Atari nieuws was, dat er een werkende en 'echte' laserprinter getoond werd, die zelfs al in Nederland zou zijn, en ook gebruikt wordt. In combinatie met de Mega ST-4 is dit in theorie een super-DTP systeem, waarbij de printer-processor in de micro zit en niet in de printer. Dit principe wordt overigens nu ook door IBM gebruikt met de Postscript-Pageprinter, die op de Efficiency VakBeurs werd geïntroduceerd. In theorie zou dit ook heel goed met de Amiga kunnen, ook daar zit een 68000 processor in, die krachtig genoeg is om de laserprinter aan te kunnen sturen. Of zoiets er zal komen is de vraag.

Op de Commodore stand in Parijs kon men al wel de Amiga versie van Word Perfect bewonderen. Het gaat hier wel om een versie 4.1 van het op PC's zo succesvolle pakket. Het biedt zowel de klassieke PC-aanpak met func tietoetsen-besturing, als een uitvoering met gebruik van de muis en rollende menu's op het scherm. Commodore kwam ook met twee nieuwe modellen PC's, waarmee de totale lijn volgens het bedrijf compleet is.



Commodore's PC-1 (gescann-de afb.)

PC-1 voor beginners

Vooral op de PC-1, de instapmachine voor MS-DOS van Commodore, hebben we lang gewacht. Geruchten, dat deze machine er aan kwam, circuleerden al geruime tijd, maar men heeft dus Parijs gekozen voor de wereldwijde primeur.

Het beginnersmodel van Commodore is nu de PC-1, een PC met de 8088 processor op een snelheid van 4,77 MHz. Het model heeft een geheugen van 512 KB en één 5,25 inch diskdrive. De PC-1 gaat in Frankrijk inclusief BTW 5.750 FF kosten (circa 2.000 gulden). Voor Nederland zijn nog geen prijzen bekend, die zullen de komende maand bepaald gaan worden. Qua uiterlijk is de PC-1 zeker een fraaie machine. Commodore liet ook het topmodel zien, de PC40/40, een van de PC40 afgeleid model met 40 MB harddisk en als kleurenkaart een EGA.

Ook in Frankrijk is nu een groot aantal nieuwe goedkope MS-DOS machine's op de markt gebracht, behalve de PC-1 van Commodore, o.a. ook de Vicki van Victor, waarvoor een gigantische reclamecampagne werd gehouden, zodat heel Parijs vol hing met posters van de Vicki. Verder krijgt de TO16 van Thompson ook veel aandacht. Het Franse bedrijf Thompson was verder de enige fabrikant die nog met 8-bits homecomputers (MO6 en TO8) op de proppen kwam.

De beginnende programmeur kan met hulp van deze serie artikelen over Basic stap voor stap leren, hoe leuk het is, om zelf meer te doen met de Commodore, dan alleen het spelen van voorgekookte spelletjes en het gebruiken van gekochte software-applicaties. Stap voor stap worden in deze cursus door Jan Bodzinga de elementaire feiten van Basic behandeld. De stof is daarbij bewust erg eenvoudig en duidelijk gehouden, zodat ook de volslagen 'leek' op programmeer-gebied erdoor kan uitgroeien tot een redelijk programmeur. Maar ook de meer gevorderde 'Basiccer' zal ongetwijfeld nog wel dingen in deze cursus tegenkomen, waarmee z'n kennis kan worden aangevuld.

Basis Basic

Deel 16 : Data-opslag

We zijn al enige tijd bezig met het opbouwen van een goed gedokumenteerd programma over een database. Dit programma zal uiteindelijk veel meer functies kunnen gaan vervullen, dan uitsluitend het vullen van de door ons in eerste instantie aangemaakte velden. Het programma is zo opgezet, dat de listing voor eigen gebruik erg gemakkelijk is aan te passen en uit te breiden. Deze keer gaan we (eindelijk) dieper in op het gebruik van externe data opslag. Het database-programma zullen we later aanvullen met de hier besproken gegevens.

Externe opslag

Als we werken met grotere bestanden dan de hoeveelheid geheugen (RAM) in onze computer toelaat, ontkomen we er niet aan, of de gegevens moeten buiten de computer worden opgeslagen, en alleen dan in het computergeheugen worden gelezen als ze nodig zijn. Om deze externe opslag te kunnen realiseren moeten we gebruik maken van een periferiek apparaat om de data te bewaren. Voor de Commodore kennen we een paar soorten en types externe opslag, waarvan de cassettetape de goedkoopste is.
Daarnaast hebben we uiteraard de floppy-diskette en eventueel de RAM-disk. Deze laatste twee zijn duurder in aanschaf en gebruik, maar bieden een groot aantal voordelen met betrekking tot snelheid en capaciteit. Om een goed gebruik te kunnen maken van deze opslag zullen we in deze aflevering ons bijna uitsluitend bezighouden met de mogelijkheden en het programmeren van opslagapparaten.

Cassettetape

De kleinere types Commodore computers werden meestal gelijk verkocht met een cassette-recorder, waarmee de programma's kunnen worden bewaard en ook databestanden kunnen worden opgeslagen. Deze recorders maken gebruik van standaard music-cassettes, die alleen qua lengte enigszins afwijken van de cassettes die gebruikt worden voor audiotoepassingen.
Dit verschil ligt voornamelijk in de onhandige manier waarmee met een te lange cassette nog kan worden gewerkt. Ook de prijs van het tapeomhulsel is inmiddels zo gering, dat het beter is om met cassettes te werken, die niet langer zijn dan zo'n 15 minuten speelduur. De opslag op een cassette gebeurd in principe op dezelfde manier als het met de floppy het geval is. Aan de hand van hele series audio-pulsjes wordt de tape beschreven met de bits, dit samen het programma of de data vormen. De computer is gelukkig in staat deze serie piepjes weer op de juiste manier terug te kunnen lezen van de tape, zodat er bij het werken met cassettes weinig gekke problemen van de computer te verwachten zijn. Het grote nadeel van tape-gebruik is de snelheid en het feit, dat er met een tape uitsluitend sprake is van sequentieel lezen en schrijven. Sequentieel wil zeggen, dat de gegevens uitsluitend **in sequentie**, ofwel achter elkaar kunnen worden gelezen. Om bijvoorbeel record-nummer 100 te kunnen lezen van een bepaald bestand, zal de cassette eerst terug moeten worden gespoeld naar het allereerste begin van het bestand.
Vanaf dat punt zal de computer het hele bestand moeten lezen, de gelezen data eventueel weer moeten weggooien, tot het gewenste record (#100) is gevonden. Deze data wordt dan ingelezen en gebruikt. Willen we daarna bijvoorbeeld record 90 inlezen, dan moeten we de hele serie handelingen weer opnieuw uitvoeren.

Om dan nog niet te spreken van de manier waarop wijzigingen in een bestaand bestand moeten worden verwerkt. De enige mogelijkheid om dit te doen is, het hele bestand succesievelijk inlezen en weer opnieuw wegschrijven op een andere tape, of achter de oude op dezelfde tape. Kortom, zeer onhandig, maar wel goedkopp en functioneel voor huis-, tuin- en keukentoepassingen.
Als we deze kwalificaties op een rij zetten, komt er bijna als vanzelfsprekend uit naar voren, dat het werken met een cassette tape bijna niet te verwezenlijken is voor bestanden die groter zijn dan dat het computergeheugen toelaat. Relatief kleine bestanden dus.

Het werken met floppy-disk kent al deze problemen niet, en verdient dan ook in de meeste gevallen de voorkeur. Maar omdat er onder ons nogal wat cassettegebruikers zijn zullen we eerst het werken met tape wat beter bekijken.

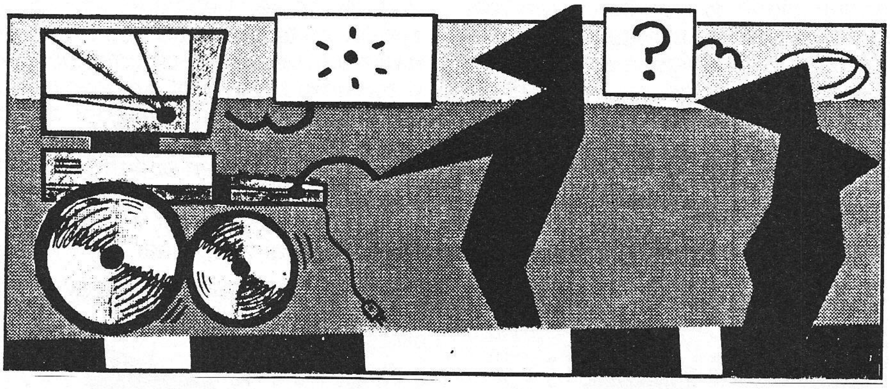
Programma's

Iedereen die in het bezit is van een computer heeft op z'n minst de basisbegrippen van het werken met een cassetterecorden onder de knie. We weten ook allemaal wel, dat het mogelijk is om programma's op tape weg te schrijven vanuit de computer, en ook dat ze weer terug kunnen worden gelezen. Vooral spelletjes worden zelfs nu nog vaak op een cassette verkocht en gebruikt. Dit zal z'n reden voornamelijk hebben in het feit, dat in doorsnee 90% van alle gebruikers nog wel beschikt over een cassetterecorder, zodat de verkochte software op cassette kan rekenen op een zo breed mogelijk verspreidingsgebied.
Om programma's te kunnen wegschrijven naar disk, zullen we eerst een programma in de computer moeten hebben. We demonstreren de beide programma-functies voor cassette aan de hand van het volgende voorbeeld.
Nadat de computer is aangezet, typen we een kort programma, dat er bijvoorbeeld als volgt uit kan zien:

```
10   REM TEST PROGRAMMA
20   FOR I = 0 to 20
30   PRINT "CASSETTE",
40   NEXT I
```

Her runnen van deze listing is nu niet belangrijk, we willen het geheel op de band wegschrijven. Daarvoor moeten we op een paar dingen letten. Als eerste is het goed om het programma vooraf nog een keer te controleren. We doen dit door het ingeven van de opdracht LIST. Daarmee krijgen we de programmaregels nogmaals op het scherm, zoals ze aanwezig zijn in de RAM van de computer. Klopt dit met de listing in het blad, of een ander zelf bedacht programma, dan kunnen we verder gaan met de SAVE-procedure. We moeten zorgen voor een tape, waarop we gegevens kunnen wegschrijven, waarbij je erom moet denken, dat de oude gegevens worden overschreven tijdens de opname.
Spoel de tapen helemaal terug tot het begin van de band. Type dan vanuit de 'READY'mode :

SAVE "TESTPRG" , 1



Hiermee vertellen we de computer, dat het in RAM aanwezige Basic-programma moet worden weggeschreven naar cassette nummer 1. Het woord SAVE komt, net als alle andere Basic-opdrachten uit het engels en betekent niets anders dan bewaren, behouden. De naam tussen de beide aanhalingstekens is de naam waarmee het programma op de tape zal komen te staan. Aan de hand hiervan is het mogelijk later het programma te kunnen herkennen. De komma en het getal 1 geven in de SAVE-opdracht aan, dat er als devicenummer 1 wordt bedoeld om de listing naar toe te zetten.
Bij Commodore kan dit in veel gevallen worden weggelaten, omdat de meeste Commodore computers als default alle SAVE en LOAD opdrachten vanaf de eerste cassettepoort interpreteren. Alleen de C-64 portable, die ooit een half jaar in Nederland op de markt is geweest wijkt hiervan af, omdat dit apparaat geen cassettepoort heeft en zelfs de routine voor het werken met cassettes mist in het ROM. Niet erg consistent van Commodore, maar het is niet anders. Sommige Commodore (business) machines beschikken over maar liefst twee cassettepoorten, waarbij device 2 als tweede tape kan worden gebruikt. Handig om gegevens te kopieren, maar wel wat ouderwets. Terug naar SAVE.
De opdracht :

SAVE "TESTPRG"

zou dus in de meeste gevallen hetzelfde resultaat hebben opgeleverd. Ook de opdracht SAVE "TESTPRG" , 1 , 0 is identiek. De laatste 0 is ook een oude optie. Deze 0 kan worden vervangen door een 2, waarbij achter het geSAVEde programma een apart EOT (end of Tape)-blok wordt geschreven, zodat de cassette hoe dan ook, stopt na het lezen van dit blok. We zullen ons met deze specifieke eigenschappen van het secundary address maar niet bezighouden. 't Is leuk om te weten dat het mogelijk is en je kunt er natuurlijk zelf mee gaan testen en proberen.

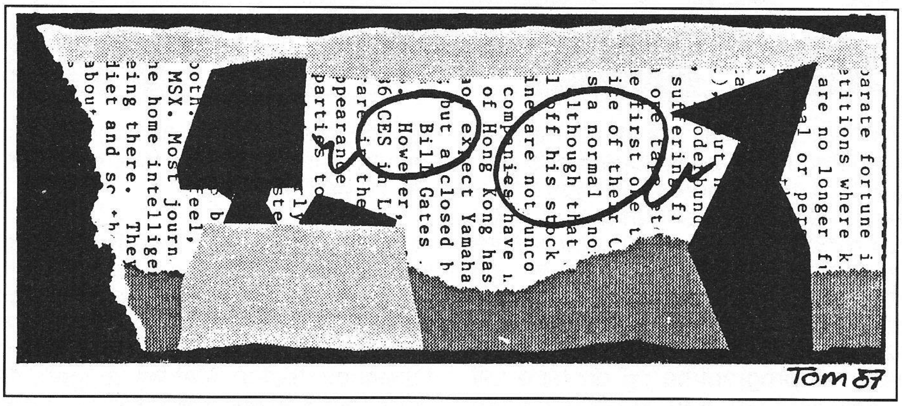
Aan de hand van de SAVE opdracht geeft de computer enige aanwijzigen, zoals de melding dat er op PLAY en RECORD moet worden gedrukt. De Commodore kan de status van een tape-toets testen. Dat wil zeggen, de computer is in staat te kijken of er een **willekeurige** toets op de recorder is ingedrukt. Druk je dus op bijvoorbeeld FFWD of PLAY in plaats van PLAY & RECORD, dan denkt de Commodore toch, dat precies gedaan is wat wordt gevraagd, en zal zondermeer de rest van de SAVE actie uitvoeren, zonder succes in dat geval. Na het indrukken van een toets verschijnt er op het scherm 'WRITING TESTPGM'. Als je weer terug met in de READY-mode, is het SAVEN klaar en staat het programma op de tape.

SAVE opnieuw

Om de test uit te breiden, kunnen we nu hetzelfde programma iets wijzigen en nogmaals wegschrijven. Als de Commodore klaar is met het SAVEN van 'TESTPGM' zal hij automatisch stoppen, terwijl toch beide toetsen PLAY & RECORD nog zijn ingedrukt. We laten dit even zo, en veranderen regel 10 in :

10 REM TEST TWEE

Daarna geven we weer de opdracht: **SAVE "TEST2".** We zien, dat nu niet meer de melding 'PRESS PLAY etc. verschijnt, maar de recorder gaat zelf lopen, en het programma wordt weggezet. Daarmee is onze SAVE-test voltooid.
Je kunt nu de band terugspoelen naar het begin van het eerste programma, waarna we zullen proberen het één en ander terug te lezen. De tijd, die nodig was om een toch erg kleine listing op tape te zetten is relatief lang, zoals je zult hebben gemerkt. Dat komt, omdat de computer voordat de echte listing wordt geschreven, eerst een paar seconden alleen de tape laat lopen voor het overslaan van eventuele aanloopstukken etc. Vervolgens wordt een zogenoemde **'header'** op de tape geschreven en daarna pas ons programma. De verhouding tussen tijd voor header en aanloop is bij korte programma's natuurlijk irreëel, maar verklaart de trage werking.



Teruglezen

Het laden van ons programma kan op verschillende manieren. Om het zo eerlijk mogelijk te testen, maken we eerst het hele computergeheugen leeg, door de opdracht NEW in te typen. Daarna controleren we of de tape aan het begin staat en typen vervolgens LOAD met RETURN.
Het scherm zal nu melden: 'PRESS PLAY ON TAPE 1' wat we zullen opvolgen. De cassette gaat lopen en na enige tijd krijgen op het scherm het woord 'LOADING', waarmee wordt aangegeven dat het (eerste) gevonden programma op de tape wordt ingelezen. Als deze actie klaar is, zal de tape stoppen. Geven we een LIST, dan zien we dat ons eerste testprogramma is ingelezen. Na de volgende LOAD-opdracht krijgen we weer 'loading...' en wordt het tweede programma geladen. Vrij gewoon dus, maar stel nu dat we meteen het tweede programma willen lezen. Hoe doen we dat?
Gelukkig kunnen we aan het LOAD-commando een naam meegeven van het gezochte programma.
In ons geval dus:

LOAD "TEST2" , 1

Na de melding van PRESS PLAY .., zien we eerst op het scherm: **found TESTPGM** en een poos later **found TEST2** met daaronder **loading**. Op deze manier kan dus selectief een programma van de tape worden gelezen. Het gebruik van de naam is daarbij een handig hulpmiddel. Zeker ook, als je wilt weten welke programmatitels op een bepaalde tape staan, zonder ze te laden. Je geeft dan een naam aan het LOAD-commando mee, waarvan je zeker bent dat deze niet op de tape voorkomt. Daardoor zal de computer de volledige tape afzoeken naar deze naam, en ondertussen alle wel gevonden namen op het scherm laten zien. Dit zal voor velen een bekend verhaal zijn, maar toch krijgen we op de redactie nog vrij veel vragen, die erop duiden dat veel lezers niet goed weten wat er allemaal met LOAD en SAVE mogelijk is.

Header

Om het niet allemaal eenvoudig te houden, hier een paar woorden over de inhoud van de 'header' op een cassetteband. Deze header bestaat uit een serie gegevens, die zeer nuttig zijn voor de interne verwerking van de computer. Voor ons echter ook aardige kennis. Er bestaan twee soorten headers, één specifiek voor programma's, de ander voor databestanden.
De headers bestaan beide uit 256 bytes, waarbij alleen de eerste bytes belangrijk zijn.
Een programma-header bevat de volgende bytes:

1 ID1 voor prg 4 voor DATA
2-3 STARTRAM startadres van programma ($803)
4-5 EINDEinde programma in RAM
6-256Programma naam met rest spaties.

Een dataheader ziet er nagenoeg hetzelfde uit, alleen de ID is verschillend en uiteraard start- en eindadressen. De machinetaal-freaks kunnen de header afzonderlijk bekijken door de routine die specifiek zoekt naar de header en deze vervolgens leest apart te activeren in machinetaal. Deze routine begint op adres $F72C voor de C-64.

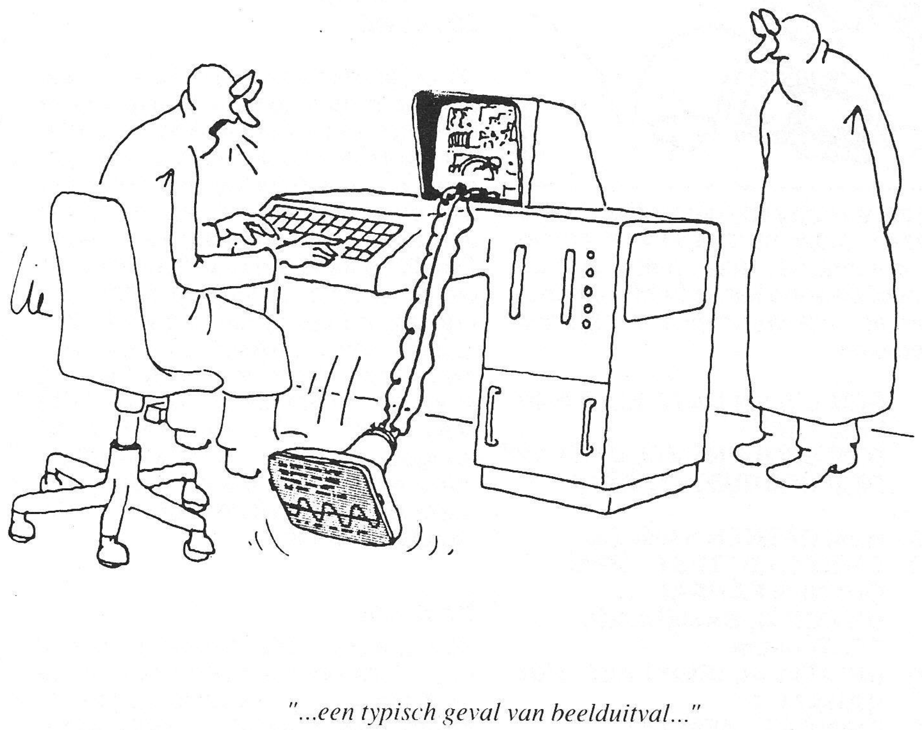
Bepaalde spelletjes gebruiken de bytes van deze header om al een stuk uit te voeren programma in te zetten, waardoor bijvoorbeeld via een interrupt muziek en tekst tijdens het laden op het beeld en SID waar te nemen valt. In het bestek van deze cursus voert uitleg hierover veel te ver.

Databestanden

Het werken met cassettetape en databestanden is niet veel anders, dan de werkwijze met Basic-programma's en tape. Dat wil zeggen wat betreft PLAY & RECORD, terugspoelen en headers.
We maken bij het wegschrijven van gegevens echter geen gebruik van de SAVE- en LOAD-opdrachten, omdat we wel met een andere manier van schrijven en lezen bezig zijn. De wijze waarop DATA en tape met elkaar communiceren kun je wat dit betreft het beste vergelijken met de manier waarop de Commodore met het videoscherm (en toetsenbord) omgaat. Het keybord en scherm van een computer kun je samen beschouwen als was het één apparaat (device). Dat komt omdat je op een toetsenbord niets als output kunt krijgen, terwijl er op een scherm weinig in te toetsen valt. Samen vormen ze echter een compleet apparaat waarmee in- en uitvoer kan worden gerealiseerd.

We weten allemaal, dat met hulp van de opdrachten PRINT, GET en INPUT van alles op het scherm kan worden gezet en vanaf het toetsenbord kan worden ingelezen. Welnu, de manier waarop dit met externe apparaten als cassetterecorder en diskdrive wordt verzorgd is in feite niet anders. Het enige verschil is het **device-nummer** dat voor ieder apparaat, dus ook het scherm en keybord, anders is.
Omdat er in de meeste gevallen communicatie tussen computer en VDU (scherm/keybord) plaatsvindt, heeft men dit device als **default** genomen, en alle print-opdrachten gaan daarom

automatisch naar device nummer 3, wat het nummer is van het videoscherm. Voor alle andere apparaten moeten we in ons programma een apart device aanspreken, en een I/O kanaal openen. Dit klinkt een heel stuk ingewikkelder dan het in werkelijkheid is, maar je moet wel eerst weten hoe het allemaal precies in z'n werk gaat. Zeker omdat het aantal kanalen wat op een bepaald moment open kan zijn, niet bij één blijft. In totaal kunnen er 16 kanalen gelijktijdig geopend zijn, die alle 16 naar verschillende devices kunnen wijzen. Zo kennen we voor de printer (device 4) een kanaal, terwijl er ook meerdere kanalen open kunnen zijn voor cassette en diskdrive. Om dit allemaal uit elkaar te kunnen houden, geven we bij het openen ieder kanaal een apart nummer of adres.
Met hulp van dit 'Logical Address' geven we later aan naar welk kanaal een bepaalde variabele moet worden geprint. Het werken met deze kanaaladressen is een goede manier om de verschillende kanalen uit elkaar te kunnen houden.
De algemene werkwijze van het versturen van berichten c.q. data naar een extern apparaat is als volgt:



"...een typisch geval van beelduitval..."

OPEN file, met kanaalnummer, apparaatnummer en secundair adres.
PRINT kanaalnummer,variabelen
CLOSE kanaalnummer

Het secundaire adres dient bij tapecommunicatie om aan te geven of er naar het bestand moet worden geschreven of teruggelezen. Bij tapegebruik kennen we daarom slechts twee secundaire adresnummers. Het getal 1 wordt gebruikt om data te schrijven, terwijl 0 als SA aangeeft, dat we willen teruglezen van een bestand.

Schrijven

Laten we maar eens een klein programma schrijven, waarmee we een serie gegevens naar de cassette wegschrijven.

```
10   REM MAKEN DATA BESTAND
     OP TAPE
20   REM SEQUENTIEEL BESTAND
30   REM MET NUMERIEKE
     VARIABELE
90   :
100  REM OPENEN KANAAL
110  OPEN 1,1,1,"TEST" :REM
     OPENEN KANAAL=1,
     DEVICE=1, SA=1(WRITE),
     TEST=naam
120  FOR I = 1 TO 256
130  PRINT I : REM PRINT OP
     SCHERM
140  PRINT#1,I :REM PRINT NAAR
     KANAAL 1
150  NEXT I
160  CLOSE 1 :REM SLUITEN
     KANAAL 1
200  END
```

Dit programma schrijft zowel op het scherm als naar de cassette 256 getallen weg. De naam van het DATA-bestand op de tape is in dit geval 'TEST'. Na de I-lus wordt in regel 160 het kanaal naar tape nummer 1 weer afgesloten. Naast de OPEN opdracht in regel 110 vinden we in regel 140 : PRINT#1,I. Deze regel, die volledig moet worden ingetypt, dus geen vraagteken (?) gebruiken voor het woord PRINT !!!, geeft aan dat de variabele I moet worden geprint naar kanaal 1, en niet naar het scherm. Zorg ervoor, dat de cassette geladen is met een nieuwe tape, en ook aan het begin staat. Na RUN krijg je al snel de melding op het scherm 'PRESS PLAY & RECORD ON TAPE 1', voor zover er inmiddels geen toetsen op de recorder waren ingedrukt. Na de CLOSE-opdracht stopt de recorder automatisch. Dan verschijnt ook READY op het scherm, om aan te geven, dat het programma is afgerond.

Lezen

Nadat we het voorgaande met succes hebben uitgevoerd, kunnen we overgaan tot het lezen van de weggeschreven getallen in file TEST. Spoel de cassette terug, en gebruik het volgende programma om de weggeschreven gegevens weer te kunnen inlezen:

```
10   REM LEZEN DATA BESTAND
     VAN TAPE
20   REM SEQUENTIEEL BESTAND
30   REM GEBRUIK NUMERIEKE
     VAR.
90   :
100  REM OPENEN KANAAL
110  OPEN 1,1,0,"TEST" :REM
     OPENEN KANAAL=1,
     DEVICE=1, SA=0(READ),
     TEST=naam
120  INPUT#1,I :REM LEES VAN
     KANAAL 1
150  PRINT I : REM ZET GELEZEN
     GETAL OP SCHERM
160  IF ST = 0 THEN 120
170  CLOSE 1 :REM SLUITEN
     KANAAL 1
200  END
```

Dit programma ziet er een beetje anders uit dan het voorgaande. Het OPEN zetten van het kanaal naar de tape verschilt in zoverre, dat het SA (secundaire adres) gewijzigd is in 0, wat aangeeft dat het bestand gebruikt wordt om uit te lezen. Zou hier op dit ogenblik een 1 gestaan hebben, dan was de melding PLAY & RECORD verschenen en zou ons vorige databestand zijn overschreven, althans de header zou vernietigd zijn en de weggeschreven data zouden daardoor nooit meer aangesproken kunnen worden. Het SA is dus bij tape-werk erg belangrijk.

In regel 120 treffen we nu een INPUT-opdracht aan, net als bij PRINT gevolgd door #1. De Basic opdracht is geen combinatie van INPUT en #1, maar luidt echt INPUT#, met daarachter het betreffende filekanaal. De gelezen waarde, in dit geval numeriek, wordt weggezet in de variabele I, waarna deze op het scherm geprint wordt. INPUT houdt automatisch op te lezen bij het vinden van een **'carriage return'** (CHR$(13)). Net als bij een normale PRINT opdracht hebben we een CHR$(13) weggeschreven bij het vorige programma, omdat we steeds een PRINT#1 hebben uitgevoerd zonder als achtervoegsel een komma of puntkomma te hebben gebruikt. Automatisch wordt dan ook naar disk en tape een carriage return geschreven. Vandaar dat de INPUT nu keurig steeds één getal naar binnen leest en op het scherm zet.

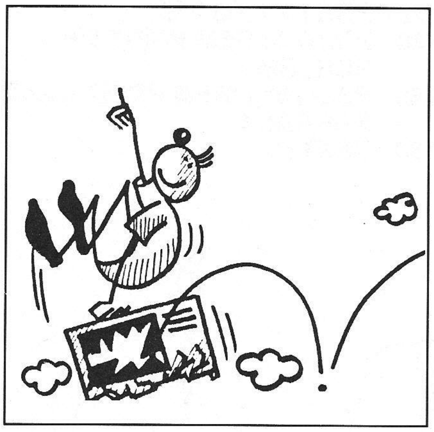
Status

Om te kunnen bekijken waar het einde van het op tape aanwezige bestand TEST zich bevindt, hebben we in ons geval verschillende mogelijkheden. We weten bijvoorbeeld, dat er precies 256 getallen op de tape staan. Daarom zouden we hier voor het inlezen ook een lus kunnen gebruiken. Maar het zal zeker niet altijd het geval zijn, dat we vooraf het aantal gegevens weten, dat in een database zit. Bovendien hadden we bijvoorbeeld als eindgetal een negatief getal kunnen wegschrijven en tijdens het inlezen iedere keer het gelezen getal hebben kunnen testen op het al of niet kleiner dan nul zijn. Maar daarmee beperken we ons tot het werken met uitsluitend positieve getallen, ook geen oplossing dus.
Nu heeft de tape (en disk) gelukkig zelf een oplossing. En dat is de status van de file/tapeunit. Deze STatus is één van de vaste, ingebouwde variabelen in de Commodore. Het is een variabele die door de programmeur alleen kan worden gelezen. Een opdracht als:

```
20    LET ST = 120
```

Dit zal dan ook altijd in een 'syntax error' eindigen. We kunnen echter wel de waarde uit ST lezen en gebruiken. ST wordt door de computer automatisch gezet, terwijl de tests op de verschillende waardes door de programmeur moeten worden gedaan. In de meeste gevallen is het zo, dat zolang ST een waarde heeft van 0, is alles met het betreffende bestand in orde. Zodra ST ongelijk is aan 0 is er iets mis. Dat is hier alles wat onthouden moet worden, hoewel ST in totaal 8 bits bezit, die ieder voor zich een bepaalde betekenis hebben, waaraan gevorderde Basic-ers kunnen zien wat er precies mis is met een file of bijvoorbeeld de printer. ST heeft dus altijd waardes die een exponent zijn van 2, bitsgewijs.
In ons lees-programma testen we slechts of ST nog altijd 0 is. Zolang dit het geval is, is alles in orde en kunnen we terug naar regel 120 om het volgende getal te kunnen lezen.
Bereikt de leeskop het einde van het bestand, waar door de Commodore automatisch een **eindblok** geschreven is tijdens het wegzetten van de data, dan wordt de variabele ST automatisch op een getal anders dan nul gezet, waardoor we in regel 170 het file afsluiten door CLOSE en ook de cassette stopt. Zoals je op het scherm kunt zien, zijn alle getallen keurig ingelezen van de tape.

Strings



We zullen hetzelfde bestand nog eens inlezen, maar nu niet in een numerieke variabele, maar met hulp van strings. Daartoe kun je het vorige programma met wat kleine wijzigingen gebruiken:

```
10    REM LEZEN DATA BESTAND
      VAN TAPE
20    REM SEQUENTIEEL BESTAND
30    REM GEBRUIK STRINGS
90    :
100   REM OPENEN KANAAL
110   OPEN 1,1,0,"TEST" :REM
      OPENEN KANAAL=1,
      DEVICE=1, SA=0(READ),
      TEST=naam
120   INPUT#1,A$ :REM LEES VAN
      KANAAL 1
150   PRINT A$ : REM ZET
      GELEZEN GETAL OP
      SCHERM
160   IF ST = 0 THEN 120
170   CLOSE 1 :REM SLUITEN
      KANAAL 1
200   END
```

Zoals je ziet, zijn de numerieke variabelen I vervangen door de stringvariabele A$. Ook op het scherm is dit te zien, door de andere layout van de getallen, die nu immers als strings worden beschouwd, waardoor er geen ruimte wordt vrijgehouden voor een plus- of min-teken etcetera. Verder is er nog weinig veranderd. Maar we zullen nu een ander bestand naar de tape zetten, waarbij er ook alfanumerieken worden weggeschreven naar bestand TEST. Daarvoor gebruiken we de volgende variant op onze listing:

```
10    REM MAKEN DATA BESTAND
      OP TAPE
20    REM SEQUENTIEEL BESTAND
30    REM STRINGS EN
      NUMERIEKEN
90    :
100   REM OPENEN KANAAL
110   OPEN 1,1,1,"TEST" :REM
      OPENEN KANAAL=1,
      DEVICE=1, SA=1(WRITE),
      TEST=naam
120   FOR I = 1 TO 256
125   A$ = CHR$(I)
130   PRINT I, A$ : REM PRINT OP
      SCHERM
140   PRINT#1,I , A$:REM PRINT
      NAAR KANAAL 1
150   NEXT I
160   CLOSE 1 :REM SLUITEN
      KANAAL 1
200   END
```

Dit programma schrijft 2 keer zoveel gegevens naar de tape. Niet alleen de lus-teller, maar ook de ASCII-waarde als karakter wordt naar de tape geschreven. Daarmee zijn we zeker van een databestand, wat een mixture is van numerieke en string-variabelen. Om dit bestand terug te lezen moeten we het laatste LEES-programma gebruiken, de listing die dus werkt met strings, want als we de numerieke versie hierop loslaten, dan krijgen we erg snel de foutmelding 'DATA MISMATCH ERROR', waarmee wordt aangeduid, dat er numerieke gegevens werden verwacht en een alfakarakter werd gelezen, die niet kan worden verwerkt.

Protocol

Bij het werken met databestanden is het in alle gevallen beter om uitsluitend met stringvariabelen de tape te beschrijven. Hoewel er ruimtebesparend kan worden gewerkt met het toepassen van numerieke variabelen is de kans op lees- en schrijffouten hoger, terwijl ook de gebruiksmogelijkheden van numerieke data uiteraard geringer zijn. Het is daarom goed als regel alleen met stringvariabelen te

werken in combinatie met data-bestanden.
Als laatste voorbeeld geven we nu nog een listing waarbij het GET# commando is toegepast:

```
10   REM LEZEN DATA BESTAND
     VAN TAPE
20   REM SEQUENTIEEL BESTAND
30   REM STRINGVARIABELEN
40   REM MET HULP VAN GET#
     OPDRACHT
90   :
100  REM OPENEN KANAAL
110  OPEN 1,1,0,"TEST" :REM
     OPENEN KANAAL=1,
     DEVICE=1, SA=0(READ),
     TEST=naam
120  GET#1,A$ :REM LEES 1
     KARAKTER VAN KANAAL 1
150  PRINT A$; : REM ZET
     GELEZEN GETAL OP
     SCHERM
160  IF ST = 0 THEN 120
170  CLOSE 1 :REM SLUITEN
     KANAAL 1
200  END
```

Ook deze listing wijkt weinig af van de vorige. Let alleen op de puntkomma achter het PRINT-commando in regel 150. Daardoor worden alle karakters achter elkaar op het scherm gezet. De 'RETURN', die ook van de tape wordt gelezen, zorgt daarbij op de juiste plaats voor een nieuwe regel. GET# wordt vaak gebruikt als er geen enkel gegeven bestaat over de manier waarop de data in het bestand is opgebouwd. Natuurlijk gaat het werken met GET# veel langzamer, omdat voor ieder karakter op de band een apart GET# moet worden uitgevoerd, kompleet met de GOTO en IF tests etcetera. Ook strings die langer zijn dan de INPUT#-opdracht toelaat (80 tekens) kunnen met hulp van GET# worden gelezen en later worden verwerkt tot records met verschillende velden of een array enzovoort.

Aan het werk

De gegevens die in dit artikel zijn verwerkt, zullen voor de meesten onder ons voldoende aanleiding zijn om de manier waarop de Commodore computer werkt met cassetterecorder en tape te kunnen begrijpen. Hoewel de werking van het STatus-byte niet volledig is vermeld, kunnen aan de hand van de gegeven voorbeelden voldoende conclusies worden getrokken om ook eigen toepassingen op tape weg te schrijven en vervolgens weer te lezen. Natuurlijk kunnen meerdere files op één tape worden weggeschreven. Het is daarbij echter nooit mogelijk om meerdere bestanden gelijktijdig op één tape te openenen. Dit heeft simpelweg te maken met de fisieke eigenschappen van een cassette. Door de sequentiele manier van lezen kan er nooit meer dan één bestand gelijktijdig langs de lees/schrijfkop van de recorder gaan, waardoor we deze beperking ontmoeten.
Bij gebruik van diskdrives wordt dit natuurlijk anders. De volgende keer zullen we verder gaan, door ook voor diskkettes dezelde tests en voorbeelden te implementeren. Daarmee kan ook ons database programma worden afgesloten.
Succes met je eigen experimenten.

Jan Bodzinga

Kleine advertenties

C-64 en diskdrive
C-64, d. drive. 1541, datarec, lichtpen, Geos 1. 3, lectuur, software, ect. Bfr14500 Tevens printer Star SG-10C. Tel. 03-88770118 (BEL)

Te koop div bladen
Computer Gazette, HCC, div. *f* 2.-/st Databecker 64 tips en tricks *f* 10,-, multi plan *f* 10,- Zaks pro Z80 *f* 15,- final cartridge2 *f* 40,- tel. 08330- 15233

Te koop C-64
c-64, d. rec, boeken, software, joystick, rasterblok *f* 700,- tel. 052-35-9447 (BEL)

JANE aangeboden
Jane voor C-128, handl, best tekstver. dbase, *f* 150. - tev KCS speedsaver 64, handl, *f* 30,- tel. 04765-2922 na 19.00 uur.

DPS 1101 printer te koop
Commodore daisywheel printer DPS 1101 *f* 3 5,- silverreed wheel pr S400 *f* 385,-. com plotter 1520 *f* 115,-final cartidge C64 *f* 50,- tel. 03487-1388

Gevraagt Astrologie
Astrologie prog. voor C-64. tel. 015-570622.

Ruilen Software
Org. software o. a. platine 64, fortran cp/m, cobol, superbase, e. a. tel. 043-477468 na 1900 uur

Scholier zoekt printer
voor normaal A4 papier. tel. 01842-1502

Programma's gevraagt
Burroughs Basic, programma's voor deze comp. Peter, kanaalln 55, 3690 bree (BEL)

Gevraagd printer
Een printer voor C-64. *f* 275,- . tel. 01110-13779

Diskdrive gezocht
D. drive voor C-64, tel. 03480-17299

Gezocht handleiding
Ned. handl van expert-cartridge ESM, tel. 09-32-03-3216883 na 1800 uur

Contact Amiga
Amiga:(int) cont. gevraagd, tel. 0032-55314755

Lichtshow gezocht
Progr. voor aansturing lichtshow, tel. 080-600184

MPS 802 printer te koop
Printer, handl, lint en Nederl. tekstverwerkingpr. TASWORD (80 Koloms) *f* 450,- tel. 03473-71751 na 17.00 uur.

Voor uw opgave: Stuur kaartje met uw tekst (max. 4 regels naar PB 112, 1260 AC Blaricum onder vermelding "Kleine advertenties CIA".

Ook dit keer weer berichter uit de wereld van ongevaarlijke computerkrakers en knappe demomakers door Darius Heydarpour.

Nieuws uit de demowereld

Vanaf nu zal ik jullie regelmatig op de hoogte houden van alle nieuwtjes uit de demowereld. De meeste informatie zal gaan in de vorm van korte berichtjes over vorm en kwaliteit van de geleverde Demo's.

De demo's die door de diverse Crackers (spelletjeskrakers) en Demomakers worden geprogrammeerd op de Commodore 64, uit pure hobby, zijn soms van zo'n goede kwaliteit dat de diverse softwarehuizen er zich nog moeilijk aan kunnen meten.

1001 Crew

Een groep die vooral kwaliteits demo's uitbrengt is *'The 1001 Crew'*. Zij waren de eerste die de welbekende 'scrollende tekst' over het gehele scherm lieten lopen.
Dus dwars door de 'zij-border'.Door een stukje uit de zij-border te 'hakken' blijft de sprite,die normaal niet te zien is, zichtbaar. Toen zij er achter kwamen, dat de gehele 'border' weggehaald kan worden programmeerden zij één van de opmerkelijkste programma's uit de geschiedenis van de Commodore 64: ESCOS (wat staat voor **E**xpanded **Sc**reen **CO**nstruction **S**et). Het produceert plaatjes die het gehele beeldscherm omvatten. Borders bestaan simpelweg niet meer. Door dit programma, wat al in 1986 is uitgebracht, is 'The 1001 Crew' een van de bekendste en zeker ook populairste groepen van de (Commodore 64) wereld.

Future Projects

Een groep die tevens van een grote populariteit mag genieten is *'Future Projects'*. Een van de bekendste programma's van hun hand is 'Future Writer'. Dit programma is een 'Letterwriter'. Dat is een programma waarin de gebruiker een brief kan typen en op disk of tape kan bewaren. Als men echter de brief weer RUNt ziet men precies wat de schrijver ervan heeft ingetypt. De cursor schrijft en verbeterd eventuele fouten. Hun nieuwste letterwriter die (hoe kon het ook anders) 'Future Writer II' heet, is de meest geavanceerde, die er op dit moment is. Men heeft de keuze uit een aantal 'Martin Galway' muziekjes en karaktersets.
Men kan de brief op schijf bewaren en hem eventueel later afmaken.Ook is het mogelijk om tijdens het schrijven van een brief de hele ingetypte tekst nog eens door te nemen en deze te onderbreken om vanaf dat punt een andere tekst te typen. Future Writer II heeft nog vele andere aspecten die deze letterwriter een uniek programma maken.

Psychos

De demo 'Psychotic 3demo' van de 'Psychos' mag ook uniek genoemd worden. Deze demo bewijst dat er nog creativiteit bestaat in demo-wereld. Na een prachtige tekening van een futuristische motor begeleidt door de welbekende scrollende tekst en een muziekje worden we in een prachtig draaiende 'kleurentunnel' gegooid. Daarna is het de beurt aan de gebruiker om de joystick ter hand te nemen en een 3 dimensionaal voorwerp te laten draaien.De scrollende tekst die onder dit hele gebeuren loopt kan ook nog aangepast worden met de + en - toets en kan zelfs terug scrollen!
De demo wordt beëindigt met een soort filmtekst.

D.H.

Infogrames brengt een schitterend vervolg-avontuur uit dat gebaseerd is op een stripverhaal, en daarvan ook de mooie graphics heeft geleend. Het wordt geleverd voor voor diverse computers, zoals de Commodore 64 en 128, de Commodore Amiga, MS-DOS, Atari ST en MSX-2.

Stripverhaal/adventure

Kinderen van de wind

Het avonturen-spel steekt zich in een nieuw jasje. Geen richtingen meer aangeven of spulletjes oppakken en plaatsen. Geen vragen meer stellen of raadsels oplossen. Deze 'adventure' ontrolt zich als een prachtig stripverhaal.

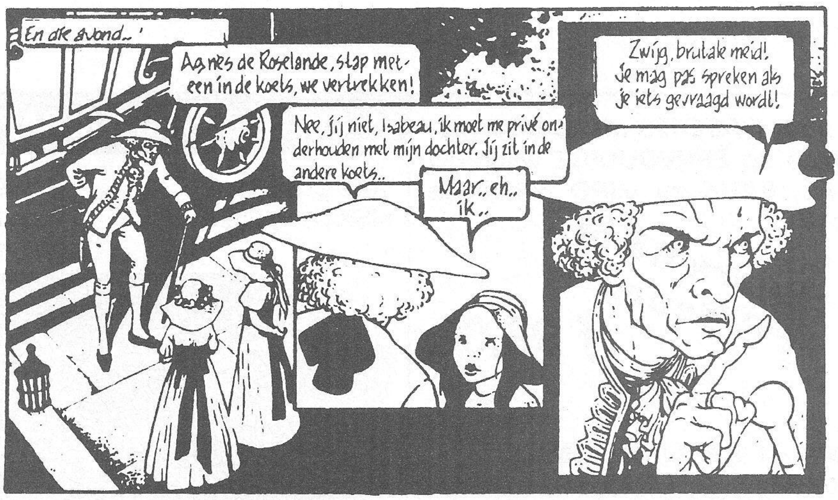
Het computer-avontuur 'Kinderen van de wind' begint waar het mooie (bijgeleverde) stripboek eindigt. Een groep mensen, door het noodlot samengebracht, gaat op zoek naar bewijsstukken om de hoofdfiguur haar identiteit terug te bezorgen. Via schitterende plaatjes wordt de speler meegesleept van de Engelse en Franse kust tot de binnenlanden van Afrika waar uiteindelijk een tipje van de sluier wordt opgelicht.

Verwisseling voor de grap

Alle problemen beginnen als de (adelijke) hoofrolspeelster acht jaar oud is. Om haar vader in de war te brengen ruilt ze voor de grap met een inwonend weesmeisje van kleding. Dat deze verwisseling het kind van de ene gruwelijke ervaring in de andere zal storten was door het achtjarige meisje niet voorzien. Door zeer realistische plaatjes in het boek, maar ook in het computer-spel, spelen we vanaf haar zestiende jaar mee in haar speurtocht naar de bewijzen van haar ware identiteit. Seks en geweld zijn regelmatig terugkerende onderdelen van haar avontuur. Maar ook humor is in dit verhaal aanwezig.

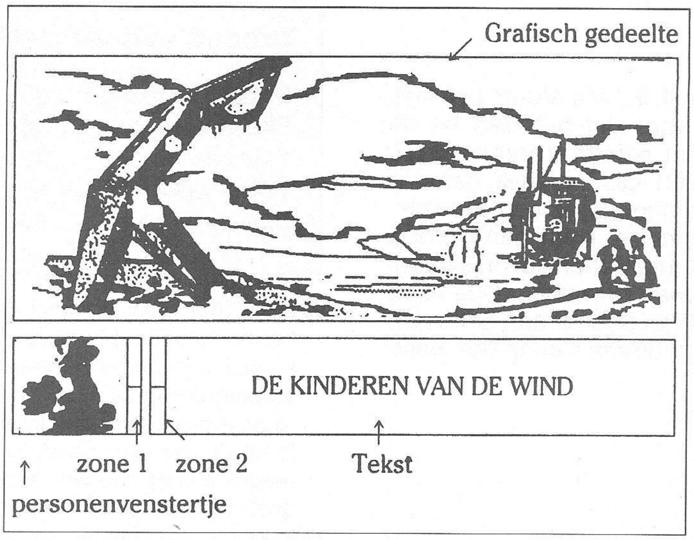
Negen situaties vol plaatjes

Via eenvoudige cursor-bewegingen laten we de medespelers aan het



woord. Een verkeerde keus uit de voorstellen die de verschillende spelers maken kan leiden tot een slechte

afloop van deze adventure. Zo zal de beginnende speler levenslang in een gevangenis kunnen belanden, verdrinken, worden gefusilleerd, de rest van zijn leven slijten in een saai Frans plaatsje, of worden aangevallen en verscheurd door een woest luipaard in Afrika.
In negen situatieplaatjes beleven we de reis van onze metgezellen. Ieder plaatje is een kort verhaal, met vele ingeprojecteerde kleinere plaatjes. Elk van de negen hoofdplaatjes kan aan het begin op een apart schijfje worden opgenomen om na een fatale beslissing niet steeds het hele verhaal opnieuw te hoeven beginnen.



Raak geschoten

Infogrames uit Frankrijk lijkt met deze vervolg-adventure raak te hebben geschoten. Ondanks de niet misselijke prijs voor dit spel vliegen ze de computerwinkels uit. Dat voor dit spel gebruik is gemaakt van tekeningen en het verhaal van de bekende striptekenaar Francois Bourgeon uit Frankrijk zal zeker een reden voor het grote succes zijn. Vaak is een zogenaamde adventure niet te genieten vanwege slechte plaatjes of een knullig verhaal. Dit programma heeft zowel uitstekende plaatjes als een mooi verhaal. De belofte dat dit avonturenspel de eerste is uit een reeks van 5 verhalen die allemaal als computerspel zullen uitkomen doet de liefhebber van adventures beslist al watertanden. De begeleidende muziek voor de MSX is van goede kwaliteit, de Commodore had helaas slechts één wijsje. Wordt de muziek voor de verschillende versies nog wat beter, dan zullen ook de volgende afleveringen van 'Kinderen van de wind' zo'n succes worden als deze eerste aflevering.

Rond honderd gulden

De adventure 'Kinderen van de wind' is verkrijgbaar voor 80 gulden voor Commodore 64 en 100 gulden betalen de bezitters van Atari ST, IBM PC en MSX2 (op diskette). De nogal groot uitgevallen verpakking was noodzakelijk om het stripverhaal te herbergen. Het spel is ook op cassette verkrijgbaar voor Commodore 64 maar ondanks de tien gulden besparing moet dat worden afgeraden. Het laden en herladen van zo'n omvangrijk programma vergt wel heel erg veel geduld van computer-avonturiers.
Kinderen van de wind werd uitgebracht door Ariola, dat inmiddels de activiteiten heeft gestaakt, Homesoft heeft de distributie overgenomen.

Rob Timmer

Print-out

Met o.a. Data-spin Amiga – Shoot'em – Yathzee – Klok, schuinschrift – Wiebel

In ons vorige nummer schreef ik: wie stuurt ons het eerste AMIGA programma, voor dat het blad bij de drukker vandaan was, lag het eerste programma bij ons in de bus. In deze PRINT-OUT rubriek daarom voor (bijna) alle commodore computers een of meerdere listings. Elders in dit nummer staat een oproep om mee te doen aan onze internationale prijsvraag. Hier zijn vele mooie prijzen mee te winnen. Dus heeft u apparatuur nodig, of mischien gebrek aan software, stuur uw eigengemaakte programma's in en wie weet.

Rob Goudriaan.

Inhoud van dit listingdeel

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd geleden heeft Jan Bodzinga hiervoor een zgn. **Checksum-programma** geschreven. Om de vele nieuwe lezers van Commodore-Info te helpen volgt hieronder nog een keer een volledige uitlegh over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of een cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap **'Fout in dataregels!'** geven dan heeft u een fout bij het overtikken gemaakt. Herstel de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap **'data is weggezet checksum testten met sys ....'** komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven. Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden (wel eerst het checksumprogramma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht **sys 49152** (C-64) of **sys 1536** (C-16 en Plus/4) in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht U het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn.

```
  1 rem *******************************
    **
  2 rem basic loader "SYNTAX.CHECKSUM"
  3 rem na de commando's 'run' en 'new'
  4 rem blijft dit programma in het ge-
  5 rem heugen. laad het te testen pro-
  6 rem gramma en tik daarna sys 49152.
  7 rem *******************************
    **
 10 i=49152  :rem beginadres
 20 reada:ifa<0then40:rem data ingeleze
    n
 30 pokei,a:i=i+1:b=b+a:goto20
 40 if b<>16844thenprint"[CLR-HOME]fout
     in dataregels!":b=0:end
 50 poke49184,148:poke49185,192
 55 i=49300
 60 read a: ifa<0then80
 70 pokei,a:b=a+b:i=i+1:goto60
 80 if b<>20068thenprint"[CLR-HOME]fout
     in dataregels! (vanaf regel 240)":
    b=0:end
 90 print"data is weggezet"
 95 print"checksum testen met sys49152"
100 data 165,43,166,44,133,163,134,164,
     169, 147
110 data 32,210,255,160,0,240,3,32,73,
     192
120 data 32,73,192,208,1,96,32,225,255,
      208
130 data 3,76,116,164,32,81,192,32,73,1
    92
140 data 240,12,201,32,240,247,24,101,1
    67,133
150 data 167,76,37,192,166,167,169,0,13
    2,168
160 data 32,205,189,169,13,32,210,255,1
    64, 168
170 data 76,17,192,200,208,2,230,164,17
    7,163
180 data 96,162,0,189,123,192,240,6,32,
    210
190 data 255,232,208,245,32,73,192,170,
    32,73
200 data 192,132,168,32,205,189,162,3,1
    69,32
210 data 32,210,255,202,208,250,169,0,1
    33,167
220 data 164,168,96,82,69,71,69,76,32,0
230 data -1
240 data 165,197,201,3,240,7,201,4,240
250 data 6,76,148,192,76,34,192,169
260 data 147,32,210,255,76,161,192
270 data -1

*** EINDE LISTING ***

    syntaxchecksum listtestprogramma

   regel 1    249
   regel 2    84
   regel 3    125
   regel 4    2
   regel 5    246
   regel 6    152
   regel 7    249
   regel 10    157
   regel 20    64
   regel 30    38
   regel 40    57
   regel 50    14
   regel 55    251
   regel 60    192
   regel 70    42
   regel 80    244
   regel 90    245
   regel 95    237
   regel 100    183
   regel 110    158
   regel 120    232
   regel 130    183
   regel 140    96
   regel 150    96
   regel 160    127
   regel 170    71
   regel 180    223
   regel 190    73
   regel 200    79
   regel 210    109
   regel 220    106
   regel 230    225
   regel 240    16
   regel 250    163
   regel 260    92
   regel 270    225

ready.
```

DATA-SPIN AMIGA

Dit eerste Amiga programma is ingezonden door P.J. Bokma uit Heerenveen. Data -Spin voor de Amiga is een eenvoudig doch overzichtelijke en gebruiksvriendelijke database voor vele toepassingen. Om er mee te werken is in principe geen manuel benodigd. Derhalve alleen een paar algemene zaken: Na het, voor de eerste keer, opstarten moet u iets invoeren. Omdat er niets op disk staat en ook niets in het geheugen aanwezig is, moet u eerst kiezen voor "0" (vanuit het hoofdmenu). U komt dan in een subdirectory: U kiest voor '3' (nieuwe file). Het programma vraagt dan hoeveel velden u wilt definiren (2-9). Gesteld u wilt een adressenbestand: U heeft dan minimaal 5 velden nodig: Naam, Adres, Postcode, Plaats en Telefoon. U toetst dus '5'. Daarna vraagt het programma die velden te benoemen. U toetst ingeval van het voorbeeld: Naam + RETURN, Adres + RETURN ect. Geen dubbele punt of statie erachter !! (Het programma zet alles netjes voor u klaar). Zodra de veldendefinitie klaar is kunt u gaan invoeren. (max. 50 tekens per regel !). Zodra u klaar bent met invoeren en dit bevestigd is door een '_' op het eerste veld in te voeren, keert het programma terug naar het hoofdmenu. U kunt het bhestand dan saven onder een eigen bedachte naam of u drukt op RETRUN. (Het programma geeft uw bestand de naam 'DATA - FILE '. Invoeren vanaf disk gaat op dezelfde wijze: alleen RETURN zal een file zoeken met de naam 'DATA - FILE '. Tenslotte nog iets over LABELS PRINTEN: Labels worden door het programma alleen geprint als er een adressenbestand gedefineerd is met als derde veld 'postcode ' of 'Postcode'. Veel plezier met Data - Spin.

```
REM Data-Spin voor Amiga c 1987 by P.J.Bokma Heeren
veen
REM -----------------------------------------------
----
CLEAR:IF FRE(x)<25000 THEN CLEAR ,125000&
av=0:sc=0:vld=0:DIM ts(200,2)
tt$="    >>>>> AMIGA ** DATA-BASE ** DATA-SPIN ** B
Y P.J.BOKMA ** JULI 1987 <<<<<"
hoofdmenu:
CLS:PRINT tt$
LOCATE 5,15:PRINT "Menu:"
LOCATE 7,15:PRINT "[0] INVOEREN"
LOCATE 9,15:PRINT "[1] BEKIJKEN"
LOCATE 11,15:PRINT "[2] ZOEKEN"
LOCATE 13,15:PRINT "[3] SORTEREN"
LOCATE 15,15:PRINT "[4] VERANDEREN"
LOCATE 7,50:PRINT "[5] PRINTEN"
LOCATE 9,50:PRINT "[6] VRIJ GEHEUGEN"
LOCATE 11,50:PRINT "[7] SAVEN"
LOCATE 13,50:PRINT "[8] WISSEN"
LOCATE 15,50:PRINT "[9] EINDE"
LOCATE 18,33:PRINT "TOETS UW KEUZE (0-9)
GOSUB keycontrol
nr=nr+1
ON nr GOTO invoeren,bekijken,zoeken,sorteren,verand
eren,printen,geheugen,saven,wissen,einde
RETURN
keycontrol:
nr$=INKEY$
IF nr$="" THEN keycontrol
nr=VAL(nr$)
IF nr=0 AND nr$<>"0" THEN BEEP:GOTO keycontrol
RETURN
toets:
nr$=INKEY$
IF nr$="" THEN toets
RETURN
netjes:
qq=21-LEN(nv$(i))
nv$(i)=nv$(i)+SPACE$(qq)+":"
RETURN
diskladen:
ON ERROR GOTO hoofdmenu
OPEN "I",#1,ip$
INPUT #1,rec:INPUT #1,vld:GOSUB dimensionering
INPUT #1,av:FOR i=1 TO av:INPUT #1,nv$(i):NEXT
INPUT #1,ar:FOR i=1 TO ar:FOR j=1 TO av
INPUT #1,re$(i,j):NEXT j:NEXT i:CLOSE #1:RETURN
diskschrijven:
sc=1:OPEN nf$ FOR OUTPUT AS #1
WRITE #1,rec:WRITE #1,vld
WRITE #1,av:FOR i=1 TO av:WRITE #1,nv$(i):NEXT
WRITE #1,ar:FOR i=1 TO ar:FOR j=1 TO av
WRITE #1,re$(i,j):NEXT j:NEXT i:CLOSE #1:RETURN
invoeren:
CLS:PRINT tt$
LOCATE 5,5:PRINT "DATA INVOER:"
LOCATE 8,30:PRINT "#1 - File laden"
LOCATE 10,30:PRINT "#2 - Toevoegen"
LOCATE 12,30:PRINT "#3 - Nieuwe file"
LOCATE 16,30:PRINT "TOETS UW KEUZE (1-3)
GOSUB keycontrol
IF nr<1 OR nr>3 THEN BEEP:GOTO hoofdmenu
ON nr GOTO fileladen,toevoegen,nieuwefile,hoofdmenu
fileladen:
CLS:PRINT tt$
LOCATE 5,5:LINE INPUT "Toets filename of druk RETUR
N:";ip$
IF LEN(ip$)<1 THEN ip$="data-file"
GOSUB diskladen:GOTO hoofdmenu
toevoegen:
IF av<2 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT "Velddefinitie:"
```

```
FOR i=1 TO av:lc=i+5:LOCATE lc,5:PRINT nv$(i):NEXT
i
LOCATE 20,5:PRINT "druk een toets voor vervolg":GOS
UB toets
invoer:
ar=ar+1:ho=0:IF ar=a1 OR FRE(x)< 500 THEN CLS:PRINT
 "     Er kunnen geen records meer bij!":FOR
x=1 TO 7500:NEXT x:GOTO hoofdmenu
CLS:PRINT "        * ";ar;" * * * * * * * * >>>>> I
NVOER <<<<< * * * * * * * * ";ar;" *"
PRINT:PRINT "    Druk . plus RETURN om te stoppen"
PRINT:FOR i=1 TO av
IF ho=1 THEN ar=ar-1:GOTO hoofdmenu
PRINT "     ";nv$(i);:LINE INPUT re$(ar,i):re$(ar,i
)=LEFT$(re$(ar,i),50)
IF LEN(re$(ar,i))=0 THEN re$(ar,i)="-"
IF re$(ar,i)="." THEN ho=1
NEXT i:GOTO invoer
nieuwefile:
IF av>1 THEN GOTO toevoegen
CLS:PRINT tt$:LOCATE 5,5:PRINT ">>>>> File definiti
e <<<<<"
PRINT:PRINT "    Toets aantal velden (2-9)";:GOSUB
keycontrol
IF nr<2 THEN hoofdmenu
PRINT nr:ar=0:IF vld=0 THEN vld=nr:GOSUB dimensione
ring:av=nr:FOR i=1 TO av
inv:
PRINT "     ";i;"# Naam: ";:LINE INPUT nv$(i)
IF nv$(i)="" THEN PRINT:PRINT "     U moet een naam
 invoeren!":PRINT:GOTO inv
nv$(i)=LEFT$(nv$(i),20):GOSUB netjes:NEXT i
PRINT:PRINT "     DEFINITIE COMPLEET ! U kunt max.
";a1-1;" records invoeren":FOR x=1 TO 7500:NE
XT x
GOTO invoer
geheugen:
CLS:PRINT tt$:LOCATE 12,5:PRINT "Er zijn nog ";FRE(
x);" bytes vrij !"
LOCATE 22,5:PRINT "druk een toets voor hoofdmenu":G
OSUB toets:GOTO hoofdmenu
einde:
IF av<1 OR ar<1 OR sc<>0 THEN SYSTEM
CLS:PRINT tt$:LOCATE 3,5:IF sc=0 THEN PRINT "Uw bes
tand is niet gesaved !"
LOCATE 5,5:PRINT "Wilt u werkelijk stoppen ? (J/N)"
GOSUB toets:IF nr$="J" OR nr$="j" THEN SYSTEM
GOTO hoofdmenu
bekijken:
IF av<2 OR ar<1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT "Tonen: Wat wilt u z
ien ?"
LOCATE 8,5:PRINT "1# - Alles":LOCATE 10,5:PRINT "2#
 - Gedeelte"
LOCATE 14,5:PRINT "Toets uw keuze (1/2)"
GOSUB keycontrol:IF nr<1 OR nr >2 THEN GOTO hoofdme
nu
IF nr=1 THEN FOR i=1 TO ar:GOSUB record:NEXT i:GOTO
 hoofdmenu
tonen:
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> Bereik tonen
<<<<<"
LOCATE 5,5:INPUT "Eerste record : ";b:LOCATE 7,5:IN
PUT "Laatste record: ";e
IF b<1 OR b>ar OR e<1 OR e>ar OR b>e THEN GOTO tone
n
FOR i=b TO e:GOSUB record:NEXT i:GOTO hoofdmenu
record:
CLS:PRINT tt$:LOCATE 3,5:PRINT "Recordnr. ";i
          LOCATE 4,5:PRINT "================":P
RINT
FOR j=1 TO av:PRINT"     ";nv$(j);" ";re$(i,j):NEXT
 j
LOCATE 22,5:PRINT "Druk een toets voor vervolg":GOS
UB toets:RETURN
zoeken:
IF av<2 OR ar<1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> ZOEKEN <<<<<"
LOCATE 5,5:PRINT "Zoeken naar: ";:LINE INPUT hu$:l=
LEN(hu$):hu=0
FOR i=1 TO ar:FOR j=1 TO av:IF hu=1 THEN zoek
IF hu$=LEFT$(re$(i,j),l) THEN hu=1
zoek:
NEXT j:IF hu=1 THEN GOSUB record:hu=0
NEXT i:GOTO hoofdmenu
sorteren:
IF av<2 OR ar<1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> SORTEREN <<<<
<":PRINT
FOR i=1 TO av:PRINT "     ";i;"# - ";nv$(i):NEXT i
PRINT:PRINT "     Op welk veld sorteren ";:GOSUB ke
ycontrol
IF nr<1 OR nr>av OR INT(av)<>av THEN GOTO hoofdmenu
vs=nr:LOCATE 22,5:PRINT "EVEN GEDULD A.U.B."
s=1:ts(1,0)=1:ts(1,1)=ar
sort1:
i=ts(s,0):r=ts(s,1):s=s-1
sort2:
j=i:k=r:s$=re$(((i+r)/2),vs)
sort3:
IF j>ar THEN sort4
IF re$(j,vs)<s$ THEN j=j+1:GOTO sort3
sort4:
IF re$(k,vs)>s$ THEN k=k-1:GOTO sort4
IF j=k THEN j=j+1:k=k-1:GOTO sort3
IF j>=k THEN sort5
```

```
IF j<k THEN FOR mo=1 TO av:te$=re$(j,mo):re$(j,mo)=
re$(k,mo):re$(k,mo)=te$
NEXT mo:j=j+1:k=k-1:GOTO sort3
sort5:
IF j<r THEN s=s+1:ts(s,0)=j:ts(s,1)=r
r=k:IF i<r THEN GOTO sort2
IF s>0 THEN GOTO sort1
GOTO hoofdmenu
veranderen:
IF av<2 OR ar<1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> WIJZIGEN <<<<
LOCATE 5,5:PRINT "1# - Bepaald nummer wijzigen"
LOCATE 7,5:PRINT "2# - Zoeken en wijzigen"
LOCATE 10,5:PRINT "Toets uw keuze (1/2)":GOSUB keyc
ontrol
IF nr<1 OR nr>2 THEN GOTO hoofdmenu
IF nr=2 THEN GOTO zenw
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> BEPAALD NUMME
R WIJZIGEN <<<<<
LOCATE 5,5:INPUT "Welk nummer wijzigen ";hi
IF hi<1 OR hi>ar OR hi<>INT(hi) THEN GOTO hoofdmenu
i=hi:GOSUB record
LOCATE 18,5:PRINT "Welk veld veranderen ? (1/";av;"
)"
GOSUB keycontrol:IF nr<1 OR nr>av THEN GOTO hoofdme
nu
LOCATE 22,5:PRINT "
    "
LOCATE 20,5:PRINT "Was  :";nv$(nr);" ";re$(i,nr)
LOCATE 21,5:PRINT "Wordt:";:LINE INPUT re$(i,nr)
IF re$(i,nr)="" THEN re$(i,nr)="-"
re$(i.nr)=LEFT$(re$(i,nr).50):GOTO hoofdmenu
zenw:
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> ZOEK EN WIJZI
G <<<<<"
hu=0:LOCATE 5,5:INPUT "Wat moet gezocht worden ";hh
$
l=LEN(hh$):FOR i=1 TO ar:FOR j=1 TO av:IF hu=1 THEN
 GOTO verder
IF hh$=LEFT$(re$(i,j),l) THEN hu=1
verder:
NEXT j:IF hu=1 THEN GOSUB record
IF hu<>1 THEN GOTO verder2
LOCATE 18,5:PRINT "Welk veld veranderen: 1/";av;" (
'v' = verder zoeken; 'h' = hoofdmenu)"
kc2:
GOSUB keycontrol2:hr=VAL(nr$):IF hr<1 OR hr>av THEN
 GOTO verder2
LOCATE 22,5:PRINT "
    "
LOCATE 20,5:PRINT "Was  :";nv$(hr);" ";re$(i,hr)
LOCATE 21,5:PRINT "Wordt:";:LINE INPUT re$(i,hr)
IF re$(i,hr)="" THEN re$(i.hr)="-"
re$(i,hr)=LEFT$(re$(i,hr),50)
verder2:
IF nr$="h" THEN GOTO hoofdmenu
hu=0:NEXT i:GOTO hoofdmenu
GOTO kc2
keycontrol2:
nr$=INKEY$:IF nr$="" THEN keycontrol2
RETURN
saven:
IF av<2 OR ar <1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> SAVEN DATA <<
<<<"
LOCATE 5,5:LINE INPUT "Toets filenaam of druk RETUR
N ";nf$
IF nf$="" THEN nf$="data-file":ON ERROR GOTO hoofdm
enu:GOSUB diskschrijven:GOTO hoofdmenu
wissen:
IF av<2 OR ar <1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> WISSEN <<<<<"
LOCATE 5,5:PRINT "1# - 1 record wissen":LOCATE 7,5:
PRINT "2# - Alle records wissen"
LOCATE 9,5:PRINT "3# - Zoek en wis"
LOCATE 13,5:PRINT "Toets uw keuze (1/3)":GOSUB keyc
ontrol:IF nr<1 OR nr >3 THEN GOTO hoofdmenu
IF nr=2 THEN CLEAR: RUN
IF nr=3 THEN GOTO zenwis
CLS:PRINT tt$:LOCATE 3,5:PRINT "Welk record wissen
 (1/";ar;")";:INPUT mo
IF mo <1 OR mo>ar OR INT(mo)<>mo THEN GOTO hoofdmen
u
PRINT:PRINT:i=mo:GOSUB record
wis:
LOCATE 22,5:PRINT "Dit record wissen (J/N) ?        "
:GOSUB toets
IF nr$<>"j" AND nr$<>"J" THEN GOTO hoofdmenu
FOR j=1 TO av:re$(i,j)=re$(ar,j):re$(ar,j)="":NEXT
j:ar=ar-1:GOTO hoofdmenu
zenwis:
CLS:PRINT tt$:LOCATE 3,5:LINE INPUT "Zoeken naar :
":hk$:hk=0:FOR i=1 TO ar:FOR j=1 TO av
IF re$(i,j)=hk$ THEN hk=1
NEXT j:IF hk=1 THEN GOSUB record:GOTO wis
hk=0:NEXT i:GOTO hoofdmenu
printen:
IF av<2 OR ar<1 THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> RECORDS PRINT
EN <<<<<"
LOCATE 5,5:PRINT "1# - Alles printen":LOCATE 7,5:PR
INT "2# - Gedeelte printen"
LOCATE 9,5:PRINT "3# - Zoek en print":LOCATE 11,5:P
RINT "4# - Labels printen (alleen voor adresb
estand!)"
LOCATE 15,5:PRINT "Toets uw keuze (1/4)":GOSUB keyc
```

```
ontrol:IF nr<1 OR nr>4 THEN GOTO hoofdmenu
ON nr GOTO aprint,gprint,zprint,labels
aprint:
b=1:e=ar:FOR i=b TO e:GOSUB routine:NEXT i:GOTO hoo
fdmenu
gprint:
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> GEDEELTE PRIN
TEN <<<<<"
LOCATE 5,5:INPUT "Eerste nummer   : ";b
LOCATE 7,5:INPUT "Laatste nummer  : ";e
FOR i=b TO e:GOSUB routine:NEXT i:GOTO hoofdmenu
zprint:
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> ZOEKEN EN PRI
NTEN <<<<<"
mp=0:LOCATE 5,5:LINE INPUT "Waar moet naar gezocht
worden ";hp$:l=LEN(hp$)
FOR i=1 TO ar:FOR j=1 TO av
IF mp=1 THEN GOTO verder3
IF hp$=LEFT$(re$(i,j),l) THEN mp=1
verder3:
NEXT j:IF mp=1 THEN GOSUB routine
mp=0:NEXT i:GOTO hoofdmenu
labels:
IF LEFT$(nv$(3),5)<> "Postc" AND LEFT$(nv$(3),5)<>
"postc" THEN GOTO hoofdmenu
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> LABELS PRINTE
N <<<<<"
LOCATE 5,5:PRINT "1# - Alles printen"
LOCATE 7,5:PRINT "2# - Gedeelte printen"
LOCATE 9,5:PRINT "3# - Zoeken en printen"
LOCATE 13,5:PRINT "Toets uw keuze (1/3)":GOSUB keyc
ontrol
IF nr<1 OR nr>3 THEN GOTO hoofdmenu
LOCATE 15,5:INPUT "Hoeveel tussenregels ":tr:IF tr=
0 THEN tr=3
ON nr GOTO alabel,glabel,zlabel
alabel:
FOR i=1 TO ar:GOSUB routine2:NEXT i:GOTO hoofdmenu
glabel:
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> GEDEELTE PRIN
TEN <<<<<"
LOCATE 5,5:INPUT "Eerste nummer  : ";b
LOCATE 7,5:INPUT "Laatste nummer : ";e
FOR i=b TO e:GOSUB routine2:NEXT i:GOTO hoofdmenu
zlabel:
CLS:PRINT tt$:LOCATE 3,5:PRINT ">>>>> ZOEKEN EN PRI
NTEN <<<<<"
mo=0:LOCATE 5,5:LINE INPUT "Waar moet naar gezocht
worden ";hl$:l=LEN(hl$)
FOR i=1 TO ar:FOR j=1 TO av
IF mo=1 THEN GOTO verder4
IF hl$=LEFT$(re$(i,j),l) THEN mo=1
verder4
NEXT j:IF mo=1 THEN GOSUB routine2
mo=0:NEXT i:GOTO hoofdmenu
routine:
ON ERROR GOTO hoofdmenu
FOR j=1 TO av:LPRINT "      ";nv$(j);re$(i,j):NEXT j
:LPRINT:RETURN
routine2:
ON ERROR GOTO hoofdmenu
FOR j=1 TO 2:LPRINT "      ";re$(i,j):NEXT j
LPRINT "      ";re$(i,3);" ";re$(i,4)
FOR uu=1 TO tr:LPRINT:NEXT uu:RETURN
dimensionering:
IF vld=2 THEN a1=1001
IF vld=3 THEN a1=801
IF vld=4 THEN a1=701
IF vld=5 THEN a1=601
IF vld=6 THEN a1=501
IF vld=7 THEN a1=401
IF vld=8 THEN a1=301
IF vld=9 THEN a1=201
DIM re$(a1,vld+1),nv$(vld+1)
RETURN
```

SHOOT'EM

Richard Tol uit Woerden is de maker van het programma Shoot'em. Het is een sterk vereenvoudigde versie van het programma West Bank. Het is de bedoeling om de bandiet neer te schieten. Op het scherm verschijnt een saloon, achter de twee onderste ramen of in de deur opening verschijnen personen. Aan u de taak om de bandiet, dit is degene met een doek voor zijn gezicht, neer te schieten. Het raken van de verkeerde personen kost een leven even als te lang wachten. Het spel bevat drie levels.

```
  1 rem shoot'em  c-64
  2 rem door richard van tol
  3 rem uit woerden
  5 hi=0
 10 rem******sprite initialisatie*****
    *
 20 p=2040:b=13
 30 for t=0 to 2
 40 pokep,b:y=b*64
 50 for x=y to y+62
 60 readz:pokex,z
 70 nextx
 80 p=p+1:b=b+1
 90 nextt
100 sc=0:li=3
110 rem score,hi-score,levens
120 poke53287,1:poke53288,1:poke53289,
    1
130 rem******programma******
140 poke53280,0:poke53281,0:print"[SHI
    FT CLR][CRSR-DOWN]";spc(12)"s[SPAC
    E]h[SPACE]o[SPACE]o[SPACE]t[SPACE]
    '[SPACE]e[SPACE]m"
```

```
150 print"[6xCRSR-DOWN][8xSPACE]welk[S
    PACE]level"
160 print"[CRSR-DOWN][12xSPACE]1=moeil
    ijk"
170 print"[CRSR-DOWN][12xSPACE]2=redel
    ijk"
180 print"[CRSR-DOWN][12xSPACE]3=makke
    lijk"
190 input le$
200 if le$<>"1" and le$<>"2" and le$<>
    "3" then 140
210 if le$="1" then le=50
220 if le$="2" then le=75
230 if le$="3" then le=100
240 gosub 640:rem saloon tekening
250 pl=int(rnd(1)*3)+1
260 no=int(rnd(1)*2)+1:poke53271,2↑no:
    poke53277,2↑no
270 poke 53269,2↑no:m=no:if no=1 then
    no=0
280 print"[HOME]score";sc;"[3xSPACE]hi
    -score";hi;"[5xSPACE]levens";li
290 if pl=1 then poke 53250+no,100
300 if pl=2 then poke 53250+no,155
310 if pl=3 then poke 53250+no,250
320 c$=""
330 poke53251+no,150
340 for t=1 to le
350 getc$:ifc$<>"" then 390
360 next
370 if m=2 then poke53269,peek(53269)a
    nd(255-2↑m):goto250:rem burger+nie
    t
380 o$="te[SPACE]laat":gosub440:goto24
    0
390 fora=0to28:poke54272+a,0:next
400 poke54277,10:poke54278,10:poke5427
    3,5:poke54276,129:poke54296,15
410 if val(c$)<>pl then o$="mis[SPACE]
    geschoten":gosub440:goto240
420 if val(c$)=pl and m=2 then o$="bur
    ger[SPACE]geraakt":gosub440:goto24
    0
430 poke53269,peek(53269)and(255-2↑m):
    sc=sc+10:goto250
440 rem******verlies leven******
450 poke53269,peek(53269) and(255-2↑m)
460 print chr$(147)
470 poke 53248,155:poke53249,120
480 poke53271,2↑0:poke53277,2↑0
490 poke53269,2↑0
500 printtab(10)o$
510 print"[16xCRSR-DOWN]";tab(14);"dru
    k[SPACE]f-1"
520 getg$:ifg$<>chr$(133) then 520
530 poke53269,peek(53269) and(255-2↑0)
540 li=li-1
550 if li=0 then goto 570:rem game ove
    r
560 return
570 rem******game over******
580 printchr$(147);"[12xSPACE]je[SPACE
    ]score[SPACE]is";sc
590 poke53269,peek(53269)and(255-2↑0)
600 if sc>hi then hi=sc
610 print"[10xCRSR-DOWN]";tab(15)"game
    [SPACE]over"
620 getg$:ifg$="" then 620
630 goto 100
640 poke53280,0:poke53281,0:poke646,1
650 printchr$(147)
660 printtab(7)"[2xSPACE]O[19xCOM Y]P"
670 printtab(7)"[2xSPACE][COM H][SPACE
    ]dead[SPACE]gulch[SPACE]saloon[SPA
    CE][COM N]"
680 printtab(7)"[2xSPACE][COM H][19xSP
    ACE][COM N]"
690 print"[4xSPACE]O[29xCOM Y]P"
700 print"[4xSPACE][COM H][SPACE]O[2xC
    OM Y]PO[2xCOM Y]P[12xSPACE]O[2xCOM
     Y]PO[2xCOM Y]P[COM N][SPACE]"
710 print"[4xSPACE][COM H][SPACE]L[2xC
    OM P][SHIFT @]L[2xCOM P][SHIFT @][
    12xSPACE]L[2xCOM P][SHIFT @]L[2xCO
    M P][SHIFT @][COM N][SPACE]"
720 print"[4xSPACE][COM H][SPACE]O[2xC
    OM Y]PO[2xCOM Y]P[12xSPACE]O[2xCOM
     Y]PO[2xCOM Y]P[COM N][SPACE]"
730 print"[4xSPACE][COM H][SPACE]L[2xC
    OM P][SHIFT @]L[2xCOM P][SHIFT @][
    12xSPACE]L[2xCOM P][SHIFT @]L[2xCO
    M P][SHIFT @][COM N][SPACE]"
740 print"[4xSPACE][COM H][29xSPACE][C
    OM N][SPACE]"
750 print"[4xSPACE][COM H][29xSPACE][C
    OM N][SPACE]"
760 print"[4xSPACE][COM H][29xSPACE][C
    OM N][SPACE]"
770 print"[4xSPACE][COM H][SPACE]O[7xC
    OM Y]P[2xSPACE]UDDDI[3xSPACE]O[7xC
    OM Y]P[COM N][SPACE]"
780 print"[4xSPACE][COM H][SPACE][COM
    H][7xSPACE][COM M][SPACE]U[5xSPACE
    ]I[2xSPACE][COM G][7xSPACE][COM M]
    [COM N][SPACE]"
790 print"[4xSPACE][COM H][SPACE][COM
    H][7xSPACE][COM M][SPACE][COM G][5
    xSPACE][COM M][2xSPACE][COM G][7xS
    PACE][COM M][COM N][SPACE]"
800 print"[4xSPACE][COM H][SPACE][COM
    H][7xSPACE][COM M][SPACE][COM G][5
    xSPACE][COM M][2xSPACE][COM G][7xS
    PACE][COM M][COM N][SPACE]"
810 print"[4xSPACE][COM H][SPACE]L[7xC
    OM P][SHIFT @][SPACE]L[5xCOM P][SH
    IFT @][2xSPACE]L[7xCOM P][SHIFT @]
    [COM N][SPACE]"
820 print"[4xSPACE][COM G][11xSPACE][C
    OM G][5xSPACE][COM M][11xSPACE][CO
    M M][SPACE]"
830 print"[4xSPACE][COM G][11xSPACE][C
    OM G][5xSPACE][COM M][11xSPACE][CO
    M M][SPACE]"
840 print"[4xSPACE][COM G][11xSPACE][C
    OM G][5xSPACE][COM M][11xSPACE][CO
    M M][SPACE]"
850 print"[40xCOM Y]"
860 return
870 rem******bang******
880 data 0,0,0,0,0,0,0,0,0
890 data 0,0,15,0,0,16,0,0,16
900 data 0,0,23,0,6,81,0,5,81
910 data 0,4,223,1,228,64,2,20,64
920 data 2,20,64,3,244,64,242,16,0
930 data 138,16,0,138,16,0,240,0,0
940 data 136,0,0,136,0,0,240,0,0
```

```
950 rem******bandiet******
960 data 0,126,0,0,255,0,0
970 data 255,0,7,255,224,1,66
980 data 128,1,24,128,0,255,0
990 data 0,126,0,7,255,224,15
1000 data 189,240,25,219,152,25,255
1010 data 152,25,255,152,25,255,152
1020 data 5,255,160,3,255,192,7
1030 data 255,224,0,0,0,0,0
1040 data 0,0,0,0,0,0,0
1050 rem******burger******
1060 data 0,255,192,0,255,192,0,255,192
1070 data 7,255,248,7,255,248,0,161,64
1080 data 0,140,64,0,128,64,0,158,64
1090 data 0,64,128,0, 63,0,0, 30,0
1100 data 3,255,224,6,127,152,6,127,152
1110 data 6,127,152,6,127,152,6,127,152
1120 data 0,0,0,0,0,0,0,0,0
```

REGEL 1	175	REGEL 450	92
REGEL 2	148	REGEL 460	77
REGEL 3	149	REGEL 470	251
REGEL 5	115	REGEL 480	234
REGEL 10	39	REGEL 490	220
REGEL 20	90	REGEL 500	57
REGEL 30	141	REGEL 510	178
REGEL 40	242	REGEL 520	141
REGEL 50	243	REGEL 530	63
REGEL 60	144	REGEL 540	184
REGEL 70	218	REGEL 550	237
REGEL 80	120	REGEL 560	142
REGEL 90	214	REGEL 570	221
REGEL 100	44	REGEL 580	68
REGEL 110	106	REGEL 590	63
REGEL 120	110	REGEL 600	227
REGEL 130	45	REGEL 610	142
REGEL 140	215	REGEL 620	113
REGEL 150	238	REGEL 630	26
REGEL 160	176	REGEL 640	243
REGEL 170	167	REGEL 650	77
REGEL 180	241	REGEL 660	20
REGEL 190	58	REGEL 670	139
REGEL 200	210	REGEL 680	62
REGEL 210	182	REGEL 690	55
REGEL 220	190	REGEL 700	111
REGEL 230	228	REGEL 710	203
REGEL 240	17	REGEL 720	111
REGEL 250	75	REGEL 730	203
REGEL 260	73	REGEL 740	59
REGEL 270	138	REGEL 750	59
REGEL 280	222	REGEL 760	59
REGEL 290	75	REGEL 770	101
REGEL 300	86	REGEL 780	128
REGEL 310	83	REGEL 790	46
REGEL 320	93	REGEL 800	46
REGEL 330	160	REGEL 810	202
REGEL 340	237	REGEL 820	117
REGEL 350	31	REGEL 830	117
REGEL 360	130	REGEL 840	117
REGEL 370	4	REGEL 850	117
REGEL 380	220	REGEL 860	142
REGEL 390	138	REGEL 870	159
REGEL 400	218	REGEL 880	147
REGEL 410	89	REGEL 890	55
REGEL 420	196	REGEL 900	69
REGEL 430	217	REGEL 910	19
REGEL 440	27	REGEL 920	71

REGEL 1030	175	REGEL 930	63
REGEL 1040	219	REGEL 940	205
REGEL 1050	78	REGEL 950	126
REGEL 1060	27	REGEL 960	176
REGEL 1070	247	REGEL 970	95
REGEL 1080	127	REGEL 980	84
REGEL 1090	164	REGEL 990	85
REGEL 1100	26	REGEL 1000	97
REGEL 1110	27	REGEL 1010	89
REGEL 1120	147	REGEL 1020	149

YATHZEE

Laurens van Kooy uit Den Haag maakte voor ons een computer versie van het overbekende dobbelspel YATHZEE. De spelregels zijn voor iedereen waarschijnlijk wel bekend. Er is geen mogelijkheid vals te spelen. De vertrouwde commodore 64 is een deskundige, onpartijdige scheidsrechter. Veel succes.

```
  1 rem ***         yahtzee          ***
  2 rem *** door:laurens van kooy ***
  3 rem ***         (c)1987          ***
100 gosub910:print"[SHIFT CLR]"tab(25)
    "[CTRL 8][CTRL 9][SPACE]f1[SPACE][
    CTRL 0]=sorteren":gosub860
110 rem *** spel ***
120 be=be+1:print"[HOME][2xCRSR-DOWN][
    COM 5]"tab(30)"beurt"be:fors=1tosp
    :gosub690
130 ford=1to5:d(d)=int(rnd(ti)*6+1):ne
    xt:p=1:mp=6:gosub590
140 gosub790:ifk$=r$thenh(p)=-h(p):ifp
    =6then160
150 gosub590:ifp=6thenprint"[HOME]"l$"
    [CTRL 9]verdergaan":goto140
160 print"[HOME]"l$"[10xSPACE]":ifp<>6
    then140
170 mp=13:ford=1to5:ifh(d)=1thend(d)=i
    nt(rnd(ti)*6+1)
180 h(d)=1:next:gosub590:poke198,0:p=0
190 p=p+1:vp=p:ifg(s,p)=1then190
200 r=1162+vp*40:poker,32:poker+1,32:r
    =1162+p*40:poker,61:poker+1,62
210 vp=p:gosub790:ifk$<>r$then200
220 rem *** punten ***
230 gosub630:pu=0:ifme=5andg(s,12)=1th
    en510
240 onpgosub350,350,350,350,350,350,38
    0,380,400,420,450,470,490
250 g(s,p)=1:tp(s)=tp(s)+pu:a=pu:gosub
    770:pu$(s,p)=b$:gosub690:tp$(s)=b$
260 ifsp<>1thenprintleft$("[COM 3]"+d$
    ,s+5);tab(13);b$:forx=1to2000:next
270 d=pl(sp(s)-1):ifsp(s)=1ora<=tp(d)+
    bo(d)then290
280 sp(s)=sp(s)-1:sp(d)=sp(d)+1:pl(sp(
    d))=d:pl(sp(s))=s:goto270
290 poke198,0:ifbe<>13ors<>spthennexts
    :gosub860:goto120
300 print"[HOME]"tab(22)"[COM 7]druk[S
    PACE]crsr[SPACE]up/down"r$;tab(22)
```

```
    "return=nieuw[SPACE]spel"
310 printtab(29)"[CTRL 8]eindstand":b=
    1:mp=sp:p=pl(1)
320 s=p:gosub690:gosub790:ifk$<>r$then
    320
330 run
340 rem *** punten berekenen ***
350 pu=a(p)*p:bh(s)=bh(s)+pu
360 ifbh(s)>=63andbn(s)=0thenbo(s)=bo(
    s)+35:bn(s)=1
370 return
380 ifme>=p-4thenpu=ao
390 return
400 ifme=3andem=2thenpu=25
410 return
420 ifa(3)>0anda(4)>0and((a(1)>0anda(2
    )>0)or(a(5)>0anda(6)>0))thenpu=30
430 ifa(2)>0anda(3)>0anda(4)>0anda(5)>
    0thenpu=30
440 return
450 ifme=1and((a(1)=1anda(6)=0)or(a(1)
    =0anda(6)=1))thenpu=40
460 return
470 ifme=5thenpu=50:ya(s)=100
480 return
490 pu=ao:return
500 rem *** yahtzee als joker ***
510 ifp>=12orp<=8thenbo(s)=bo(s)+ya(s)
    :goto240
520 ifg(s,d(1))=0thenvp=p:p=d(1):goto2
    00
530 bo(s)=bo(s)+ya(s):pu=val(mid$("253
    040",(p-9)*2+1,2)):goto250
540 rem *** stenen sorteren ***
550 forz=1to4:fory=1to4
560 ifd(y)>d(y+1)thend=d(y+1):d(y+1)=d
    (y):d(y)=d:h=h(y+1):h(y+1)=h(y):h(
    y)=h
570 nexty,z:return
580 rem *** stenen printen ***
590 ford=1to5:ifh(d)=-1thenprint"[COM
    3]";
600 ifd=pandmp=6thenprint"[CTRL 9]";
610 printleft$(d$,d*5-4)d$(d(d));"[CTR
    L 2][CTRL 0]";:next:return
620 rem *** stenen tellen ***
630 gosub550:me=0:em=0:ao=0:ford=1to6:
    a(d)=0:next
640 ford=1to5:a(d(d))=a(d(d))+1:ao=ao+
    d(d):next
650 ford=1to6:ifa(d)>emthenem=a(d)
660 ifa(d)>methenem=me:me=a(d)
670 next:return
680 rem *** score blad ***
690 print"[HOME]"tab(18)"[2xCRSR-DOWN]
    [COM 5]speler"s:printtab(17)"[COM
    4][COM A]CCCCC[COM 3]score[SPACE]b
    lad[COM 4]CCCCC[COM S]"
700 form=1to13:printtab(17)g$(g(s,m));
    left$("[6xCTRL 6][COM 6]",m);m$(m)
    ;"[CTRL 8]";
710 printtab(35);pu$(s,m):next:printta
    b(17)"[COM Q]CCCCCCCCCCCCCCCCCCCC[
    COM W]"
720 a=bh(s):gosub770:printtab(17)"B[CT
    RL 6]bovenste[SPACE]helft.:[SPACE]
    "b$
730 a=tp(s)-bh(s):gosub770:printtab(17
    )"B[COM 6]onderste[SPACE]helft.:[S
    PACE]"b$
740 a=bo(s):gosub770:printtab(17)"B[CO
    M 7]bonus[SPACE]totaal...:[SPACE]"
    b$
750 a=tp(s)+bo(s):gosub770:printtab(17
    )"B[CTRL 8]generaal[SPACE]totaal:[
    SPACE]"b$
760 printtab(17)"[COM Z]CCCCCCCCCCCCCC
    CCCCCC[COM X][CTRL 2]":return
770 b$=right$("[SPACE]"+str$(a),3)+"[C
    OM 4]B":return
780 rem *** besturing ***
790 getk$:ifk$=""then790
800 ifk$="[CRSR-DOWN]"thenp=p+1:ifp=mp
    +1thenp=1
810 ifk$="[CRSR-UP]"thenp=p-1:ifp=0the
    np=mp
820 ifk$="[F1]"andb<>1thengosub550:gos
    ub590
830 ifmp=13andg(s,p)=1thengoto800
840 return
850 rem *** socore bord ***
860 ifsp=1thenreturn
870 print"[HOME][2xCRSR-DOWN]"l$"[COM
    4][COM A][COM 5]spe[COM 4][COM R][
    CTRL 6]p[COM 4][COM R][CTRL 8]pnt[
    COM 4][COM S]"r$;l$"[COM Q]CCC[SHI
    FT +]C[SHIFT +]CCC[COM W]"
880 ford=1tosp:printl$"B[COM 5]"n$(d)"
    [COM 4]B[CTRL 6]"chr$(sp(d)+48)"[C
    OM 4]B[CTRL 8]"tp$(d):next
890 printl$"[COM Z]CCC[COM E]C[COM E]C
    CC[COM X]":return
900 rem *** initialiseren ***
910 dimg(6,13),pu$(6,13),m$(13),d$(13)
    :g$(0)="B[CTRL 8][2xSPACE]":g$(1)=
    "B[COM 3]--"
920 poke53281,0:poke53280,0:d$="[HOME]
    [20xCRSR-DOWN]"
930 forx=1to6:fory=1to13:pu$(x,y)="[3x
    SPACE][COM 4]B":next:sp(x)=x:pl(x)
    =x:h(x)=1
940 tp$(x)="[CTRL 8][2xSPACE]0[COM 4]B
    ":next:forx=1to13:readm$(x):next:l
    $="[6xCRSR-RIGHT]":r$=chr$(13)
950 a$="UCCCI[CRSR-DOWN][5xCRSR-LEFT]B
    ":c$="B[CRSR-DOWN][5xCRSR-LEFT]JCC
    CK":b$="B[CRSR-DOWN][5xCRSR-LEFT]B
    "
960 forx=0to6:a$(x)=mid$("[3xSPACE]Q[3
    xSPACE]Q[3xSPACE]QQ[SPACE]QQQ[SPAC
    E]QQQ",x*3+1,3):next
970 forx=1to13:reada,b,c:d$(x)=a$+a$(a
    )+b$+a$(b)+b$+a$(c)+c$:next
980 print"[SHIFT CLR][CTRL 2]":forx=1t
    o150
990 printleft$(d$,rnd(1)*20+1);tab(rnd
    (ti)*35);d$(rnd(ti)*6+1);:next
1000 forx=0to6:printleft$(d$,11)"[COM 6
    ]"tab(x*5+2);d$(x+7):next:forx=1to
    5000:next
1010 print"[SHIFT CLR]"tab(16)"[CTRL 8]
    yahtzee"tab(56)"-------"
1020 print"[CRSR-DOWN][COM 6]-de[SPACE]
    spelregels[SPACE]van[SPACE]deze[SP
    ACE]yahtzee[SPACE]versie"
1030 print"[SPACE]zijn[SPACE]precies[SP
    ACE]dezelfde[SPACE]als[SPACE]die[S
```

```
     PACE]van[SPACE]de"
1040 print"[SPACE]originele[SPACE]versi
     e.[SPACE]het[SPACE]programma[SPACE
     ]kent"
1050 print"[SPACE]ook[CRSR-RIGHT]de[SPA
     CE]yahtzee[SPACE]als[SPACE]joker[S
     PACE]en[SPACE]bonus[7xSPACE]punten
     "
1060 print"[CRSR-DOWN][CTRL 6]-de[SPACE
     ]bediening[SPACE]gaat[SPACE]met[SP
     ACE]de[SPACE]crsr[SPACE]up/donw"
1070 print"[SPACE]en[SPACE]de[SPACE]<re
     turn>[SPACE]toetsen."
1080 print"[SPACE][CTRL 8][CTRL 9][SPAC
     E]f1[SPACE][CTRL 0][CTRL 6][SPACE]
     is[SPACE]om[SPACE]de[SPACE]dobbels
     tenen[SPACE]op[SPACE]volgorde[2xSP
     ACE]te[SPACE]leggen."
1090 print"[SPACE][CTRL 2]witte[CTRL 6]
     [SPACE]stenen[SPACE]worden[SPACE]o
     ver[SPACE]geworpen."
1100 print"[SPACE][COM 3]rode[CTRL 6][2
     xSPACE]stenen[SPACE]worden[SPACE]v
     astgehouden."
1110 print"[SPACE]de[SPACE]steen[SPACE]
     die[SPACE][CTRL 2][CTRL 9]reverse[
     CTRL 0][CTRL 6][SPACE]is[SPACE]afg
     edrukt[SPACE]is[SPACE]de[SPACE]cur
     sor."
1120 print"[2xCRSR-DOWN][COM 5]-met[SPA
     CE]hoeveel"r$"[SPACE]spelers[SPACE
     ]wil"r$"[SPACE]je[SPACE]spelen[SPA
     CE]?":p=1:b=1:mp=6
1130 printtab(24)"[3xCRSR-UP][COM 7]geb
     ruik"r$;tab(23)"de[SPACE]cursor"r$
     ;tab(21)"up/down-toets[CRSR-DOWN]"
1140 printtab(14)"[CTRL 2][5xCRSR-UP]"d
     $(p):gosub790:sp=p:ifk$<>r$then114
     0
1150 print"[SHIFT CLR][COM 7][CRSR-DOWN
     ]-voer[SPACE]nu[SPACE]de[SPACE]ini
     tialen[SPACE]van[SPACE]de[SPACE]sp
     elers[SPACE]in[6xSPACE](max[SPACE]
     3[SPACE]letters)"
1160 fors=1tosp:print"[HOME][5xCRSR-DOW
     N][COM 5]initialen[SPACE]speler[SP
     ACE][2xCRSR-UP][CTRL 2]"d$(s)"[COM
      5][2xCRSR-UP][3xSPACE]sp"chr$(s+4
     8)
1170 printtab(23);"[CRSR-UP]";:inputn$:
     n$(s)=left$(n$+"[3xSPACE]",3):next
     :b=0:return
5000 rem *** data ***
5010 dataenen,tweeen,drieen,vieren,vijv
     en,zessen,three of a kind,carre
5020 datafull house,kleine straat,grote
      straat,yahtzee,chance
5030 data0,2,0,3,0,1,1,2,3,4,0,4,4,2,4,
     4,4,4,4,2,2,2,6,4,4,6,4,6,2,2,6,2,
     6
5040 data6,5,6,6,5,6

     ** EINDE LISTING yahtzee
```

```
REGEL 1     165
REGEL 2     58
REGEL 3     248
REGEL 100   21
REGEL 110   191
REGEL 120   153
REGEL 130   255
REGEL 140   154
REGEL 150   188
REGEL 160   143
REGEL 170   145
REGEL 180   28
REGEL 190   95
REGEL 200   203
REGEL 210   86
REGEL 220   101
REGEL 230   110
REGEL 240   63
REGEL 250   171
REGEL 260   138
REGEL 270   150
REGEL 280   119
REGEL 290   234
REGEL 300   120
REGEL 310   49
REGEL 320   108
REGEL 330   138
REGEL 340   244
REGEL 350   204
REGEL 360   112
REGEL 370   142
REGEL 380   61
REGEL 390   142
REGEL 400   140
REGEL 410   142
REGEL 420   50
REGEL 430   147
REGEL 440   142
REGEL 450   178
REGEL 460   142
REGEL 470   34
REGEL 480   142
REGEL 490   175
REGEL 500   0
REGEL 510   66
REGEL 520   240
REGEL 530   90
REGEL 540   202
REGEL 550   101
REGEL 560   236
REGEL 570   41
REGEL 580   120
REGEL 590   6
REGEL 600   214
REGEL 610   177
REGEL 620   28
REGEL 630   153
REGEL 640   105
REGEL 650   33
REGEL 660   117
REGEL 670   74
REGEL 680   26
REGEL 690   219
REGEL 700   224
REGEL 710   143
REGEL 720   88
REGEL 730   196
REGEL 740   170
REGEL 750   206
REGEL 760   132
REGEL 770   228
REGEL 780   62
REGEL 790   129
REGEL 800   238
REGEL 810   147
REGEL 820   46
REGEL 830   255
REGEL 840   142
REGEL 850   125
REGEL 860   70
REGEL 870   48
REGEL 880   76
REGEL 890   54
REGEL 900   91
REGEL 910   214
REGEL 920   36
REGEL 930   248
REGEL 940   11
REGEL 950   187
REGEL 960   113
REGEL 970   216
REGEL 980   165
REGEL 990   129
REGEL 1000  57
REGEL 1010  85
REGEL 1020  40
REGEL 1030  134
REGEL 1040  48
REGEL 1050  3
REGEL 1060  25
REGEL 1070  163
REGEL 1080  100
REGEL 1090  250
REGEL 1100  18
REGEL 1110  4
REGEL 1120  80
REGEL 1130  239
REGEL 1140  178
REGEL 1150  253
REGEL 1160  210
REGEL 1170  2
REGEL 5000  165
REGEL 5010  176
REGEL 5020  125
REGEL 5030  151
REGEL 5040  161

READY.
```

TELEX-WRITER

Telex-writer is een programma van Roland Huijbregts uit Terheijden. Het programma is gebaseerd op het bekende real-time writers. Alle toetsaanslagen worden in het geheugen opgeslagen op de manier waarop zij zijn ingetypt. Op een door uw zelf gekozen snelheid. De tekst kan ook op een cassette of diskette worden weggeschreven. Een ideale manier om uw eigen levende brieven te schrijven. Een gesavede tekst kan zonder problemen uit elk hoofdprogramma worden opgeroepen d.m.v. sys 8013. De afdruksnelheid kan worden ingesteld door poke 8042,x waarbij x de waarde kan krijgen tussen 1 en 255. Verdere informatie in het programma.

```
 10 rem telex-writer / 64
 20 rem door r.huijbregts
 30 rem uit terheijden -- 1987
 40 rem
 50 goto590:rem data lezen
 60 poke53281,246:poke53280,254:printc
    hr$(147)
 70 print"[CTRL 1][HOME][40xSHIFT *]";
 80 print"[CTRL 2][CTRL 9][13xSPACE]te
    lex-writer[15xSPACE][CTRL O]";
 90 print"[CTRL 1][40xSHIFT *]";
100 print"[CTRL 2][CTRL 9][8xSPACE]doo
    r[SPACE]r.huijbregts[15xSPACE][CTR
    L O]";
110 print"[CTRL 1][40xSHIFT *]";
120 print"[CTRL 2]keuze-menu:"
130 print"[2xCRSR-DOWN][12xCRSR-RIGHT]
    1.[SPACE]uitleg[SPACE]telex-writer
    "
140 print"[2xCRSR-DOWN][12xCRSR-RIGHT]
    2.[SPACE]invoeren"
150 print"[2xCRSR-DOWN][12xCRSR-RIGHT]
    3.[SPACE]bekijken"
160 print"[2xCRSR-DOWN][12xCRSR-RIGHT]
    4.[SPACE]wegschrijven"
170 print"[2xCRSR-DOWN][12xCRSR-RIGHT]
    5.[SPACE]stoppen"
180 print"[2xCRSR-DOWN][2xCRSR-RIGHT]u
    w[SPACE]keuze[SPACE]is[SPACE]:";
190 inputa:ifa<1anda>5then190
200 onagosub220,330,420,480,580
210 goto60
220 printchr$(147):print"[CTRL 1][10xS
    PACE]uitleg[SPACE]telex-writer[8xS
    PACE]"
230 print"[CTRL 2][2xCRSR-DOWN][2xCRSR
    -RIGHT]telex-writer[SPACE]is[SPACE
    ]een[SPACE]programma[SPACE]om"
240 print"[2xCRSR-RIGHT][CRSR-DOWN]tek
    sten[SPACE]letter[SPACE]voor[SPACE
    ]letter[SPACE]in[SPACE]te"
250 print"[CRSR-DOWN][2xCRSR-RIGHT]hal
    en[SPACE]en[SPACE]ze[SPACE]later[S
    PACE]als[SPACE]bijvoorbeeld"
260 print"[2xCRSR-RIGHT][CRSR-DOWN]het
    [SPACE]begin-scherm[SPACE]van[SPAC
    E]een[SPACE]programma"
```

```
270 print"[2xCRSR-RIGHT][CRSR-DOWN]wee
    r[SPACE]letter[SPACE]voor[SPACE]le
    tter[SPACE]af[SPACE]te[SPACE]druk-
    "
280 print"[2xCRSR-RIGHT][CRSR-DOWN]ken
    .
290 print"[CRSR-DOWN][2xCRSR-RIGHT]het
    [SPACE]lijkt[SPACE]op[SPACE]een[SP
    ACE]telex-apparaat[SPACE]waar"
300 print"[2xCRSR-RIGHT][CRSR-DOWN]de[
    SPACE]tekst[SPACE]letter[SPACE]voo
    r[SPACE]letter[SPACE]wordt[SPACE]a
    f-"
310 print"[CRSR-DOWN][2xCRSR-RIGHT]ged
    rukt[SPACE]op[SPACE]het[SPACE]papi
    er.
320 print"[CRSR-DOWN][2xCRSR-RIGHT]dru
    k[SPACE]<return>";:inputa$:return
330 printchr$(147):print"[CTRL 1][2xCR
    SR-DOWN][2xCRSR-RIGHT]er[SPACE]zij
    n[SPACE]een[SPACE]aantal[SPACE]fun
    cties:"
340 print"[CTRL 2][3xCRSR-DOWN][2xCRSR
    -RIGHT]f1:[SPACE]scherm[SPACE]scho
    on"
350 print"[2xCRSR-DOWN][2xCRSR-RIGHT]f
    2:[SPACE]borderkleur[SPACE]met[SPA
    CE]1[SPACE]verhogen"
360 print"[2xCRSR-DOWN][2xCRSR-RIGHT]f
    3:[SPACE]schermkleur[SPACE]met[SPA
    CE]1[SPACE]verhogen"
370 print"[2xCRSR-DOWN][2xCRSR-RIGHT]f
    4:[SPACE]stoppen[SPACE]met[SPACE]d
    e[SPACE]invoer[SPACE]"
380 print"[2xCRSR-DOWN][2xCRSR-RIGHT]n
    .b.:deze[SPACE]functies[SPACE]zijn
    [SPACE]alleen[SPACE]te"
390 print"[7xCRSR-RIGHT]gebruiken[SPAC
    E]tijdens[SPACE]het[SPACE]invoeren
    ."
400 print"[CRSR-DOWN][2xCRSR-RIGHT]dru
    k[SPACE]<return>[SPACE]om[SPACE]te
    [SPACE]starten";:inputa$:
410 poke53280,254:poke53281,246:print"
    [COM 7]":sys49152:return
420 print"[SHIFT CLR][HOME][2xCRSR-DOW
    N][16xCRSR-RIGHT][CTRL 1]bekijken"
430 print"[CTRL 2][3xCRSR-DOWN][2xCRSR
    -RIGHT][CTRL 2]de[SPACE]snelheid[S
    PACE]van[SPACE]afdrukken."
440 print"[2xCRSR-RIGHT][2xCRSR-DOWN]n
    ormaal[SPACE]is[SPACE]dat[SPACE]64
    ,
450 input"[2xCRSR-RIGHT][2xCRSR-DOWN]v
    ul[SPACE]nu[SPACE]het[SPACE]getal[
    SPACE]in[SPACE](1/255):";n
460 ifn<0orn>255thengoto450
470 poke53280,254:poke53281,246:print"
    [COM 7]":poke8042,n:sys8013:return
480 printchr$(147):print"[HOME][CTRL 1
    ][2xCRSR-DOWN][12xCRSR-RIGHT]wegsc
    hrijven"
490 print"[2xCRSR-DOWN][CTRL 2][2xCRSR
    -RIGHT]disk[SPACE]of[SPACE]cassett
    e[SPACE](d/c):"
500 getde$:ifde$<>"d"andde$<>"c"then50
    0
510 ifde$="d"thenpoke49420,8
520 ifde$="c"thenpoke49420,1
530 print"[2xCRSR-DOWN][2xCRSR-RIGHT]n
    aam[SPACE]programma(max16):"
540 input"[2xCRSR-RIGHT]";r$:ifr$=""th
    en540
550 q=len(r$):poke49413,q:t=0
560 fori=qto1step-1:c$=right$(r$,i):x=
    asc(c$):poke49461+t,x:t=t+1:next
570 sys49408:return
580 sys64738
590 print"[SHIFT CLR][8xCRSR-DOWN][10x
    CRSR-RIGHT]even[SPACE]wachten[SPAC
    E]a.u.b."
600 fort=49152to49407:reada:poket,a:ne
    xt
610 fort=7968to8192:reada:poket,a:next
620 fort=49408to49460:reada:poket,a:ne
    xt
630 goto60
800 data162,0,160,32,134,251,132,252
810 data169,64,141,138,2,234,234,234
820 data234,234,169,147,32,210,255,32
830 data221,192,234,234,234,32,228,255
840 data201,0,240,246,234,234,234,32
850 data240,192,201,133,208,13,160,0
860 data145,251,230,251,208,220,230,25
    2
870 data76,18,192,201,134,208,16,238
880 data82,208,160,0,145,251,230,251
890 data208,205,230,252,76,23,192,201
900 data135,208,16,238,33,208,160,0
910 data145,251,230,251,208,185,230,25
    2
920 data76,23,192,201,136,208,26,160
930 data0,145,251,230,251,208,9,230
940 data252,160,0,169,0,145,251,96
950 data160,0,169,0,145,251,96,234
960 data234,201,17,208,16,160,0,145
970 data251,32,210,255,230,251,208,135
980 data230,252,76,23,192,201,29,240
990 data236,201,145,240,232,201,157,24
    0
1000 data228,234,234,234,160,0,145,251
1010 data230,251,208,2,230,252,32,210
1020 data255,162,15,142,24,212,169,129
1030 data141,4,212,169,15,141,5,212
1040 data169,170,141,0,212,141,1,212
1050 data202,208,232,162,0,142,4,212
1060 data142,5,212,76,23,192,234,234
1070 data234,234,234,234,234,72,168,72
1080 data164,211,177,209,133,249,9,128
1090 data145,209,104,168,104,96,234,234
1100 data72,168,72,164,211,165,249,145
1110 data209,104,168,104,96,234,234,234
1500 data234,234,169,147,32,210,255,76
1510 data93,31,234,234,72,168,72,164
1520 data211,177,209,133,249,9,128,145
1530 data209,104,168,104,96,234,234,72
1540 data168,72,164,211,165,249,145,209
1550 data104,168,104,96,234,169,147,32
1560 data210,255,162,0,160,32,134,251
1570 data132,252,32,44,31,160,0,177
1580 data251,162,0,160,0,200,208,253
1590 data232,224,64,208,248,32,63,31
1600 data201,133,208,10,230,251,208,170
1610 data230,252,76,34,31,234,201,134
1620 data208,13,238,32,208,230,251,208
1630 data209,230,252,76,90,31,234,201
```

```
1640 data135,208,7,238,33,208,76,133
1650 data31,234,201,136,208,5,76,133
1660 data31,234,234,201,32,208,8,32
1670 data210,255,76,133,31,234,234,201
1680 data15,240,244,201,17,240,240,201
1690 data29,240,236,201,145,240,232,201
1700 data157,240,228,201,0,208,4,96
1710 data234,234,234,32,210,255,230,251
1720 data208,2,230,252,162,15,142,24
1730 data212,169,129,141,4,212,169,15
1740 data141,5,212,169,170,141,0,212
1750 data141,1,212,202,208,232,169,0
1760 data141,0,212,141,1,212,141,24
1770 data212,76,90,31,234,234,234,234
1780 data234
1900 data162,53,160,193,169,1,32,189
1910 data255,169,1,162,8,160,255,32
1920 data186,255,165,251,133,253,165,25
     2
1930 data133,254,230,253,208,2,230,254
1940 data169,32,133,251,169,31,133,252
1950 data169,251,166,253,164,254,32,216
1960 data255,96,234,234,234
```

** EINDE LISTING telex-writer

REGEL 10	180	REGEL 430	215
REGEL 20	58	REGEL 440	44
REGEL 30	150	REGEL 450	180
REGEL 40	143	REGEL 460	208
REGEL 50	136	REGEL 470	50
REGEL 60	131	REGEL 480	20
REGEL 70	187	REGEL 490	155
REGEL 80	77	REGEL 500	47
REGEL 90	168	REGEL 510	23
REGEL 100	108	REGEL 520	15
REGEL 110	168	REGEL 530	212
REGEL 120	2	REGEL 540	37
REGEL 130	16	REGEL 550	80
REGEL 140	33	REGEL 560	199
REGEL 150	255	REGEL 570	111
REGEL 160	76	REGEL 580	170
REGEL 170	231	REGEL 590	180
REGEL 180	122	REGEL 600	136
REGEL 190	199	REGEL 610	45
REGEL 200	8	REGEL 620	139
REGEL 210	239	REGEL 630	239
REGEL 220	74	REGEL 800	219
REGEL 230	128	REGEL 810	240
REGEL 240	92	REGEL 820	30
REGEL 250	145	REGEL 830	80
REGEL 260	192	REGEL 840	220
REGEL 270	196	REGEL 850	216
REGEL 280	18	REGEL 860	114
REGEL 290	34	REGEL 870	242
REGEL 300	102	REGEL 880	220
REGEL 310	139	REGEL 890	23
REGEL 320	223	REGEL 900	181
REGEL 330	70	REGEL 910	124
REGEL 340	140	REGEL 920	235
REGEL 350	161	REGEL 930	174
REGEL 360	166	REGEL 940	136
REGEL 370	88	REGEL 950	136
REGEL 380	109	REGEL 960	173
REGEL 390	202	REGEL 970	67
REGEL 400	167	REGEL 980	231
REGEL 410	82	REGEL 990	114
REGEL 420	72	REGEL 1000	23

REGEL 1010	214	REGEL 1650	180
REGEL 1020	32	REGEL 1660	124
REGEL 1030	124	REGEL 1670	19
REGEL 1040	166	REGEL 1680	7
REGEL 1050	165	REGEL 1690	64
REGEL 1060	184	REGEL 1700	134
REGEL 1070	37	REGEL 1710	67
REGEL 1080	42	REGEL 1720	173
REGEL 1090	86	REGEL 1730	237
REGEL 1100	44	REGEL 1740	170
REGEL 1110	85	REGEL 1750	168
REGEL 1500	38	REGEL 1760	106
REGEL 1510	197	REGEL 1770	234
REGEL 1520	41	REGEL 1780	28
REGEL 1530	37	REGEL 1900	196
REGEL 1540	94	REGEL 1910	141
REGEL 1550	41	REGEL 1920	140
REGEL 1560	219	REGEL 1930	20
REGEL 1570	125	REGEL 1940	31
REGEL 1580	165	REGEL 1950	90
REGEL 1590	186	REGEL 1960	9
REGEL 1600	59		
REGEL 1610	225		
REGEL 1620	24	READY.	
REGEL 1630	230		
REGEL 1640	194		

DIGI-ONTWERPER

Van dezelfde maker als het voorgaande programma is het programma digi ontwerper. Als u beide programma's met elkaar vergelijkt ziet u wat voor verschillende programma's een programmeur kan maken. Door middel van digitale programmering kan het geluid via de sidchip veel meer golven aannemen dan de vier reeds bekende. Zelfs spraakachtige tonen zijn nu mogelijk. De golf wordt op het scherm zichtbaar als staafdiagram. Er gaat een wereld van geluiden voor u open.

```
  1 rem **  digi golfontwerper   **
  2 rem door    erik de neve
  3 rem         uit oostzaan
  4 rem voor      c= info
  5 rem
100 poke650,128:poke56,159:poke55,0:cl
    r
110 fora=0to51:readd:pokea+40908,d:nex
    t
120 dimw%(9,33):c=5:print"[SHIFT CLR]"
    spc(48)"-[SPACE]6[SPACE]seconden[S
    PACE]geduld[SPACE]-"
130 fora=1to32:w%(6,a)=sin(a*.198)*7.6
    +8:next
140 fora=1to32:w%(5,33-a)=sin((a-1)*.1
    96*4)*(7.5-a/32*7.5)+7.5:next
150 fora=1to16:w%(7,a/2)=15:w%(8,a)=a/
    16*15:w%(9,a)=w%(8,a):next
160 fora=17to32:w%(7,a)=w%(7,a-16):w%(
    8,a)=w%(9,33-a):w%(9,a)=(a-16)/16*
    15:next
```

```
170 print"[CTRL 1][SHIFT CLR]"chr$(142
    )chr$(8)
180 poke53280,11:poke53281,12:gosub750
190 printspc(13)"[3xCRSR-DOWN][CTRL 2]
    [CTRL 9][3xSPACE]menu[3xSPACE][CRS
    R-DOWN][CTRL 1]"
200 print"[CRSR-DOWN][2xSPACE][f1][3xS
    PACE]inleiding"
210 print"[CRSR-DOWN][2xSPACE][f3][3xS
    PACE]ontwerpen"
220 print"[CRSR-DOWN][2xSPACE][f5][3xS
    PACE]golflengte[SPACE](£)"
230 print"[CRSR-DOWN][2xSPACE][f7][3xS
    PACE]digitaal[SPACE]geluid[SPACE]a
    an
240 print"[CRSR-DOWN][2xSPACE][[SPACE]
    ←][3xSPACE]digitaal[SPACE]geluid[S
    PACE]uit
250 gosub690:gosub720:onagoto270,460,6
    50
260 gosub710:goto250
270 gosub750:print"[2xCRSR-DOWN][CTRL
    2][SPACE]door[SPACE]middel[SPACE]v
    an[SPACE]digitale[SPACE]programmer
    ing[2xSPACE]kan[SPACE]het[SPACE]ge
    luid";
280 print"[SPACE]via[SPACE]de[SPACE]si
    d-chip[SPACE]veel[5xSPACE]meer[SPA
    CE]golfvormen[SPACE]aannemen[SPACE
    ]dan[SPACE]de"
290 print"[SPACE]vier[SPACE]reeds[SPAC
    E]bekende.":print"[SPACE]zelfs[SPA
    CE]spraak-achtige[SPACE]tonen[SPAC
    E]zijn[SPACE]nu"
300 print"[SPACE]mogelijk."
310 print"[SPACE]de[SPACE]golf[SPACE]w
    ordt[SPACE]bij[SPACE]het[SPACE]ont
    werpen[SPACE]zicht-[2xSPACE]baar[S
    PACE]als[SPACE]een[SPACE]staafgraf
    iek,";
320 print"[SPACE]die[SPACE]te[7xSPACE]
    veranderen[SPACE]is[SPACE]d.m.v.[S
    PACE]de[SPACE]cursortoetsen."
330 print"[SPACE]het[SPACE]resultaat[S
    PACE]kan[SPACE]beluisterd[SPACE]wo
    rden[SPACE]met[SPACE]een[SPACE]dru
    k[SPACE]op[SPACE]de[SPACE]spatieba
    lk,";
340 print"[SPACE]en[SPACE]de[SPACE]gol
    f-[2xSPACE]lengte[SPACE](dus[SPACE
    ]ook[SPACE]de[SPACE]frequentie)[SP
    ACE]varieert[SPACE]met[SPACE]de[SP
    ACE]+/-";
350 print"toetsen."
360 gosub820:gosub750:print"[CTRL 2][2
    xCRSR-DOWN][SPACE]de[SPACE]golven[
    SPACE]5-9[SPACE]zijn[SPACE]aldus"
370 print"[SPACE]voorgeprogrammeerd:"
380 print"[CRSR-DOWN][SPACE]5[SPACE]←[
    SPACE]anharmonische[SPACE][o]-klan
    k
390 print"[SPACE]6[SPACE]←[SPACE]harmo
    nische[SPACE]sinusgolf"
400 print"[SPACE]7[SPACE]←[SPACE]blok/
    pulsgolf
410 print"[SPACE]8[SPACE]←[SPACE]drieh
    oek/triangelgolf
420 print"[SPACE]9[SPACE]←[SPACE]zaagt
    andgolf
430 print"[CRSR-DOWN][SPACE]deze[SPACE
    ]zijn[SPACE]naar[SPACE]eigen[SPACE
    ]inzicht[SPACE]te"
440 print"[SPACE]veranderen;[SPACE]op[
    SPACE]0-4[SPACE]kun[SPACE]u[SPACE]
    zelf":print"[SPACE]ontwerpen."
450 gosub820:goto170
460 i=1:h=1:print"[CTRL 2][SHIFT CLR]"
470 print"[HOME][CTRL 1]"
480 fora=1to4:print"[CTRL 9][SPACE]E":
    forb=1to3:print"[CTRL 9][SPACE]'":
    nextb,a:print"[CTRL 9][2xSPACE]";
490 fora=1to4:print"[CTRL 9]Z'''''''";
    :next:gosub810:print
500 print"[CRSR-DOWN][CTRL 2][CTRL 9]c
    rsr[CTRL 1][CTRL 0]=ontwerpen[3xSP
    ACE][CTRL 2][CTRL 9]spatie[CTRL 1]
    [CTRL 0]=toon[SPACE]uit/aan"
510 print"[CTRL 2][CTRL 9]0-9[SPACE][C
    TRL 1][CTRL 0]=golf[SPACE]nummer[S
    PACE][CTRL 2][CTRL 9][SPACE]+[SPAC
    E]-[2xSPACE][CTRL 1][CTRL 0]=golfl
    engte"
520 printspc(17)"[CTRL 2][CTRL 9][3xSP
    ACE]m[2xSPACE][CTRL 1][CTRL 0]=naa
    r[SPACE]menu":print"[CTRL 2][SPACE
    ]nr.[SPACE]="c
530 print"[SPACE]£[3xSPACE]="peek(4093
    4)"[CRSR-LEFT][SPACE]"
540 fora=1to32:forb=1tow%(c,a)+1:poke1
    705+a-b*40,160:nextb,a
550 gosub690
560 gosub780:w%=w%(c,h):ifa=145andw%<1
    5thenw%(c,h)=w%+1:gosub800:goto560
570 ifa=17andw%>0thenw%(c,h)=w%-1:gosu
    b800:goto560
580 ifa=157andh>1thenh=h-1:gosub810:go
    to560
590 ifa=29andh<32thenh=h+1:gosub810:go
    to560
600 ifa=32thengosub710:goto550
610 ifa>47anda<58thenc=a-48:goto460
620 ifa=77goto170
630 i=peek(40934)+(a=45)-(a=43):ifi<25
    6andi>0thenpoke40934,i:print"[SPAC
    E][CRSR-UP]£[3xSPACE]="i"[CRSR-LEF
    T][SPACE]"
640 goto560
650 print"[SHIFT CLR][CTRL 1]":gosub75
    0:print"[CTRL 2][2xCRSR-DOWN][2xSP
    ACE]£[SPACE]="peek(40934)
660 input"[CRSR-DOWN][2xSPACE]nieuwe[S
    PACE]golflengte[SPACE](1-255)[3xCR
    SR-RIGHT][COM B][3xCRSR-LEFT]";a
670 ifa>0anda<256thenpoke40934,a
680 goto 170
690 geta$:ifa$<>""then690
700 return
710 fora=1to32:poke831+a,w%(c,a):next:
    sys40908:return
720 geta$:ifa$=""then720
730 a=asc(a$):ifa<133ora>136then720
740 a=a-132:return
750 print"[SHIFT CLR][CTRL 1][CRSR-DOW
    N]";:gosub760:printspc(10)"[CTRL 2
    ]digi[SPACE]golfontwerper[CTRL 1]"
    :gosub760:return
```

```
760 fora=0to9:print"CDEEEDCFRRRF";:nex
    t
770 return
780 geta$:ifa$=""then780
790 a=asc(a$):return
800 b=w%(c,h):poke1625+h-(b-1)*40,160:
    poke1625+h-b*40,32:return
810 poke1745+h,30:poke1746+h,32:poke17
    44+h,32:return
820 poke198,0:wait198,1:return
830 data 120,169,0,141,3,220,141,17,20
    8
840 data 173,1,220,133,255,162,0,189,6
    4
850 data 3,141,32,208,141,24,212,160,3
    4
860 data 136,208,253,232,138,41,31,170
870 data 172,1,220,196,255,240,230,169
880 data 27,141,32,208,141,17,208,88,9
    6

    ** EINDE LISTING digi ontwerp
```

REGEL 1	66
REGEL 2	165
REGEL 3	240
REGEL 4	129
REGEL 5	143
REGEL 100	109
REGEL 110	160
REGEL 120	63
REGEL 130	188
REGEL 140	205
REGEL 150	164
REGEL 160	140
REGEL 170	255
REGEL 180	237
REGEL 190	48
REGEL 200	176
REGEL 210	225
REGEL 220	181
REGEL 230	202
REGEL 240	206
REGEL 250	71
REGEL 260	127
REGEL 270	252
REGEL 280	106
REGEL 290	131
REGEL 300	93
REGEL 310	226
REGEL 320	246
REGEL 330	114
REGEL 340	71
REGEL 350	45
REGEL 360	107
REGEL 370	110
REGEL 380	197
REGEL 390	93
REGEL 400	20
REGEL 410	74
REGEL 420	197
REGEL 430	141
REGEL 440	173
REGEL 450	130
REGEL 460	64
REGEL 470	128
REGEL 480	65

REGEL 490	187
REGEL 500	205
REGEL 510	154
REGEL 520	152
REGEL 530	110
REGEL 540	62
REGEL 550	44
REGEL 560	159
REGEL 570	228
REGEL 580	119
REGEL 590	122
REGEL 600	12
REGEL 610	73
REGEL 620	13
REGEL 630	24
REGEL 640	36
REGEL 650	81
REGEL 660	66
REGEL 670	156
REGEL 680	33
REGEL 690	30
REGEL 700	142
REGEL 710	16
REGEL 720	102
REGEL 730	59
REGEL 740	61
REGEL 750	195
REGEL 760	238
REGEL 770	142
REGEL 780	108
REGEL 790	55
REGEL 800	221
REGEL 810	251
REGEL 820	40
REGEL 830	59
REGEL 840	75
REGEL 850	58
REGEL 860	25
REGEL 870	29
REGEL 880	88

READY.

LIBELLE-ANIMATIE

Het programma van E.J.G. de Neve uit Oostzaan is bedoeld om het kille computerscherm op te fleuren met een zomers stukje natuur. Een gemoedelijk zwermpje libelles fladderen niets vermoedend tussen alle basic-regels door. Nadat we onze 64 op deze manier tot biocomputer hebben gepromofeerd, kunnen we onder toeziend oog van de 8 insekten verder werken in basic.

```
  0 rem   *** libelle animatie ***
  1 rem             door
  2 rem        e.j.g. de neve
  3 rem          uit oostzaan
  4 rem
100 print"[SHIFT CLR][CRSR-DOWN][SPACE
    ]geduld......[6xSPACE]"
110 fora=49152to49152+264:readd:pokea,
    d:next
120 print"[SHIFT CLR][CTRL 2][7xSPACE]
    ***[SPACE]libelle[SPACE]animatie[S
    PACE]***[SPACE]"
130 print"[CRSR-DOWN][8xSPACE]oproepen
    [SPACE]met[SPACE]sys[SPACE]49152"
140 print"[10xSPACE]interrupt[SPACE]pr
    ogramma,"
150 print"[5xSPACE]dus[SPACE]werkt[SPA
    CE]tegelijk[SPACE]met[SPACE]basic.
    "
160 print"[CRSR-DOWN][5xSPACE]haal[SPA
    CE]de[SPACE]natuur[SPACE]in[SPACE]
    uw[SPACE]c-64[SPACE]!"
170 print"[5xSPACE]druk[SPACE]een[SPAC
    E]toets[SPACE]en[SPACE]let[SPACE]o
    p..."
180 poke198,0:wait198,1:sys49152
190 fort=1to3000:next
200 poke198,0:print"[9xCRSR-DOWN]":end
210 data 120, 76, 143, 192, 120, 198,
    251, 208, 19, 169, 3, 133
220 data 251, 234, 234, 162, 7, 189, 2
    48, 7, 73, 1, 157, 248
230 data 7, 202, 16, 245, 162, 15, 160
    , 128, 140, 18, 212, 173
240 data 27, 212, 41, 1, 208, 6, 254,
    0, 208, 76, 51, 192
250 data 222, 0, 208, 202, 16, 237, 17
    3, 31, 208, 208, 225, 76
260 data 49, 234, 0, 0, 2, 0, 0, 207,
    0, 4, 223, 173
270 data 110, 159, 85, 84, 34, 0, 0, 4
    1, 0, 0, 136, 64
280 data 0, 16, 0, 20, 0, 0, 42, 0, 0,
    84, 0, 0
290 data 170, 0, 0, 84, 0, 0, 168, 0,
    1, 80, 0, 0
300 data 160, 0, 1, 64, 0, 2, 128, 0,
    1, 0, 0, 2
310 data 0, 0, 207, 0, 4, 219, 173, 11
    0, 157, 85, 84, 34
320 data 0, 0, 41, 0, 0, 136, 64, 0, 1
    6, 0, 255, 120
```

```
330 data 169, 192, 141, 21, 3, 169, 4,
     141, 20, 3, 169, 255
340 data 141, 21, 208, 169, 128, 160,
     7, 153, 248, 7, 136, 16
350 data 250, 160, 160, 169, 0, 234, 1
     53, 255, 31, 136, 208, 247
360 data 160, 23, 185, 63, 192, 153, 4
     1, 32, 136, 208, 247, 160
370 data 57, 185, 85, 192, 153, 72, 32
     , 136, 208, 247, 160, 0
380 data 140, 16, 208, 160, 15, 169, 1
     28, 153, 0, 208, 136, 16
390 data 250, 169, 255, 141, 14, 212,
     141, 15, 212, 160, 7, 185
400 data 1, 193, 153, 39, 208, 136, 16
     , 247, 169, 0, 141, 33
410 data 208, 169, 0, 141, 32, 208, 16
     9, 15, 141, 134, 2, 169
420 data 1, 133, 251, 88, 96, 1, 7, 12
     , 13, 15, 10, 14, 1

** EINDE LISTING
```

REGEL 0	204	REGEL 270	1
REGEL 1	195	REGEL 280	130
REGEL 2	166	REGEL 290	243
REGEL 3	240	REGEL 300	185
REGEL 4	143	REGEL 310	106
REGEL 100	74	REGEL 320	134
REGEL 110	171	REGEL 330	10
REGEL 120	178	REGEL 340	114
REGEL 130	64	REGEL 350	255
REGEL 140	124	REGEL 360	209
REGEL 150	27	REGEL 370	121
REGEL 160	170	REGEL 380	153
REGEL 170	27	REGEL 390	199
REGEL 180	61	REGEL 400	62
REGEL 190	219	REGEL 410	108
REGEL 200	255	REGEL 420	94
REGEL 210	214		
REGEL 220	78		
REGEL 230	147	READY.	
REGEL 240	159		
REGEL 250	151		
REGEL 260	142		

KLOK

Een aantal maanden terug hadden we een programma border klokje. Hierop maakte W.C.F. de Bont een variatie. De tijd in beeld kan worden aan- en uitgezet met SHIFT-CTRL.

```
100 rem
110 print"[SHIFT CLR][3xCRSR-DOWN][3xC
     RSR-RIGHT]even[SPACE]geduld"
120 j=49152:fori=0to168:readd:pokej+i,
     d:next
130 input"uren:[4xSPACE]";u$
140 input"minuten:[SPACE]";m$
150 input"seconden:";s$
160 pokej,val(left$(u$,1))+48:pokej+1,
     val(right$(u$,1))+48
170 pokej+3,val(left$(m$,1))+48:pokej+
     4,val(right$(m$,1))+48
180 pokej+6,val(left$(s$,1))+48:pokej+
     7,val(right$(s$,1))+48
190 sys49283
200 print"aan-[SPACE]en[SPACE]uitzette
     n[SPACE]met[SPACE]'shift-ctrl'"
1000 data 48,48,46,48,48,32,48,48,32,48
     ,48,1,234,234,234,234
1010 data 169,15,157,29,216,189,0,192,1
     57,29,4,202,16,242,76,49
1020 data 234,162,10,254,0,192,189,0,19
     2,201,58,208,99,169,48,157
1030 data 0,192,16,66,169,58,141,5,192,
     16,6,208,211,240,223,234
1040 data 234,202,254,0,192,189,0,192,2
     01,54,208,68,169,48,157,0
1050 data 192,224,0,240,201,202,202,224
     ,1,208,200,238,1,192,173,1
1060 data 192,201,52,208,196,173,0,192,
     201,50,208,189,169,48,141,1
1070 data 192,141,0,192,16,168,165,2,23
     4,105,32,133,2,16,181,169
1080 data 32,16,179,120,169,33,162,192,
     141,20,3,142,21,3,88,96
1090 data 173,141,2,201,5,208,12,169,1,
     77,11,192,141,11,192,208
1100 data 154,240,154,173,11,192,16,247
     ,0

** EINDE LISTING klok
```

REGEL 100	143	REGEL 1020	52
REGEL 110	221	REGEL 1030	187
REGEL 120	105	REGEL 1040	246
REGEL 130	241	REGEL 1050	247
REGEL 140	207	REGEL 1060	85
REGEL 150	4	REGEL 1070	231
REGEL 160	102	REGEL 1080	137
REGEL 170	54	REGEL 1090	170
REGEL 180	72	REGEL 1100	122
REGEL 190	168		
REGEL 200	77		
REGEL 1000	64	READY.	
REGEL 1010	213		

SCHUINSCHRIFT

Van W.C.F. de Bont kregen we een aantal programma's toegestuurd. Een van de programma's is schuinschrift. Een leuk programma dat verder geen uitleg nodig heeft. En kan zonder verdere handelingen direkt gerunt worden. (wel natuurlijk eerst saven).

```
10 rem italics / commodore-64
20 rem door helmut de bont
30 rem reusel 04976 - 2583
40 rem
50 rem
60 print"[SHIFT CLR][2xCRSR-DOWN][10x
    CRSR-RIGHT]even[SPACE]geduld[SPACE
    ]a.u.b..."
100 for l=49152 to 49239
110 read a$
120 s = asc(left$(a$,1)):if s>64 then
    s=s-7
```

```
130 s = (s-48) * 16
140 u = asc(right$(a$,1)):if u>64 then
     u=u-7
150 u = u-48+s:poke l,u:i=i+u:next
160 if i<>9832 then print"fout[SPACE]i
     n[SPACE]dataregels!":stop
170 printchr$(147)
180 sys 49197
190 :
200 data ea,ea,ea,ea,ea,a9,01,8d
210 data 19,d0,ad,16,d0,49,01,8d
220 data 16,d0,a5,02,c9,f6,f0,0b
230 data 18,69,04,85,02,8d,12,d0
240 data 4c,81,ea,a9,32,85,02,8d
250 data 12,d0,4c,31,ea,78,a9,32
260 data 85,02,8d,12,d0,ad,11,d0
270 data 29,7f,8d,11,d0,a9,01,8d
280 data 0d,dc,8d,1a,d0,a9,00,a2
290 data c0,8d,14,03,8e,15,03,58
300 data 60,c0,8d,14,03,8e,15,03

   ** EINDE LISTING schuinschrift
```

REGEL 10	3	REGEL 190	58
REGEL 20	78	REGEL 200	172
REGEL 30	104	REGEL 210	63
REGEL 40	143	REGEL 220	74
REGEL 50	143	REGEL 230	21
REGEL 60	85	REGEL 240	71
REGEL 100	51	REGEL 250	61
REGEL 110	236	REGEL 260	52
REGEL 120	11	REGEL 270	72
REGEL 130	211	REGEL 280	97
REGEL 140	20	REGEL 290	33
REGEL 150	71	REGEL 300	26
REGEL 160	46		
REGEL 170	77		
REGEL 180	172	READY.	

WIEBEL

Het spel wiebel, ook van Richard Tol, is duidelijk gebaseerd op frogger. De bedoeling is Wiebel over de weg te sturen zonder dat hij (of is het zij) geraakt wordt. Na 10 keer krijg je een aantal bonuspunten en loopt Wiebel beduidend minder snel.

```
 1 rem wiebel / commodore-64
 2 rem door richard van tol
 3 rem uit woerden
10 hi=0
20 x=53248:y=53249:kl=53287:o=53269:p
   =2040:x1=150:y1=210:le=20:pu=0:wi=
   4
30 poke53280,0:poke53281,0:poke646,1
40 pokep,13
50 forc=1to5:pokep+c,14:next
60 s1=13*64:s2=14*64:v=0
70 forc=s1tos1+62:readz:pokec,z:next
80 forc=s2tos2+31:readz:pokec,z:next
85 forc=928to958:pokec,0:next
90 forc=0to5:pokekl+c,1:next
100 poke53271,62:poke53277,62
110 pokex,150:pokey,210
120 a$="RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
    R"
130 b$="FF[4xSPACE]FF[4xSPACE]FF[4xSPA
    CE]FF[4xSPACE]FF[4xSPACE]F"
140 x2=0:x3=255:x4=0:x5=255:x6=0
150 y2=180:y3=155:y4=130:y5=105:y6=80
160 pokeo,63
170 print"[SHIFT CLR][3xCRSR-DOWN]";a$
180 print"[6xCRSR-DOWN]";b$
190 print"[8xCRSR-DOWN]";a$
200 print"[HOME][2xCRSR-DOWN]"
210 forc=1to18:printtab(31)"[COM K]":n
    ext
220 pokeo,63:pokex,150:pokey,210
230 print"[HOME]levens";wi"[SPACE]punt
    en";pu;"[3xSPACE]hi-score";hi
240 x2=x2+10:x3=x3-17:x4=x4+5:x5=x5-9:
    x6=x6+18
250 ifx2>250 then x2=0
260 ifx3<0 then x3=255
270 ifx4>250 then x4=0
280 ifx5<0 then x5=255
290 ifx6>250 then x6=0
300 pokex,x1:pokex+2,x2:pokex+4,x3:pok
    ex+6,x4:pokex+8,x5:pokex+10,x6
310 pokey,y1:pokey+2,y2:pokey+4,y3:pok
    ey+6,y4:pokey+8,y5:pokey+10,y6
320 j=peek(56320)
330 ifj=119 then x1=x1+le:pokex,x1
340 ifj=123 then x1=x1-le:pokex,x1
350 ifj=125 then y1=y1+le:pokey,y1
360 ifj=126 then y1=y1-le:pokey,y1
370 bo=peek(53278)
```

```
380 if bo=3 or bo=5 or bo=9 or bo=17 o
    r bo=33 then goto530
390 if y1<=50 then v=v+1:pu=pu+10:x1=1
    50:y1=210:goto230
400 if v=10 then 620
410 goto230
420 data 0,0,0,0,0,0,0,4,64
430 data 0,2,128,0,2,128,0,3,128
440 data 14,7,192,14,13,224,15,255,176
450 data 7,255,120,0,28,120,0,57,248
460 data 0,7,244,0,63,236,0,15,216
470 data 0,71,244,0,127,224,0,63,192
480 data 0,70,192,0,28,112,0,124,60
490 data 60,60,0,60,60,0,255,254,0
500 data 128,3,192,128,3,224,128,3,224
510 data 128,3,224,128,3,192,255,254,0
520 data 60,60,0,60,60
530 x1=150:y1=210:pokex,x1:pokey,y1
540 forc=1to240:poke53281,c:next
550 poke53278,0:wi=wi-1
560 ifwi<0 then580
570 goto220
580 poke53269,0:print"[10xCRSR-DOWN]";
    tab(14);"[CTRL 9][CTRL 2][SPACE]ga
    me[SPACE]over[SPACE]"
590 ifpu>hi then hi=pu
600 getc$:ifc$="" then 600
610 restore:goto20
620 pokeo,1:poke53271,1:poke53277,1
630 pokey,150:pokex,150
640 print"[SHIFT CLR][7xCRSR-DOWN][SPA
    CE]je[SPACE]hebt[SPACE]wiebel[SPAC
    E]over[SPACE]de[SPACE]10[SPACE]weg
    en[SPACE]gezet."
650 print"[SPACE]kan[SPACE]je[SPACE]he
    t[SPACE]ook[SPACE]met[SPACE]een[SP
    ACE]tragere[SPACE]wiebel."
660 print"[SPACE]je[SPACE]krijgt[SPACE
    ]250[SPACE]ptn[SPACE]en[SPACE]een[
    SPACE]extra[SPACE]leven!!"
670 wi=wi+1:pu=pu+250:c$=""
680 getc$:ifc$="" then 680
690 le=le-3:ifle<9 then le=9
700 v=0:goto100

    ** EINDE LISTING wiebel
```

REGEL 1	178	REGEL 160	123
REGEL 2	148	REGEL 170	67
REGEL 3	149	REGEL 180	228
REGEL 10	115	REGEL 190	5
REGEL 20	162	REGEL 200	18
REGEL 30	243	REGEL 210	88
REGEL 40	119	REGEL 220	79
REGEL 50	219	REGEL 230	27
REGEL 60	14	REGEL 240	206
REGEL 70	165	REGEL 250	112
REGEL 80	163	REGEL 260	121
REGEL 85	143	REGEL 270	116
REGEL 90	237	REGEL 280	125
REGEL 100	154	REGEL 290	120
REGEL 110	154	REGEL 300	144
REGEL 120	201	REGEL 310	156
REGEL 130	222	REGEL 320	15
REGEL 140	230	REGEL 330	166
REGEL 150	233	REGEL 340	162

REGEL 350	167	REGEL 570	29
REGEL 360	169	REGEL 580	21
REGEL 370	95	REGEL 590	1
REGEL 380	209	REGEL 600	103
REGEL 390	108	REGEL 610	177
REGEL 400	51	REGEL 620	169
REGEL 410	30	REGEL 630	157
REGEL 420	209	REGEL 640	154
REGEL 430	219	REGEL 650	190
REGEL 440	28	REGEL 660	107
REGEL 450	176	REGEL 670	219
REGEL 460	71	REGEL 680	111
REGEL 470	172	REGEL 690	23
REGEL 480	113	REGEL 700	140
REGEL 490	66		
REGEL 500	25		
REGEL 510	26	READY.	
REGEL 520	251		
REGEL 530	152		
REGEL 540	224		
REGEL 550	4		
REGEL 560	82		

PICCOLO C-16

In dit spel van Peter Boersma uit Stadskanaal bent u de piccolo die de post naar kamer 0 moet brengen. Nu is dat in normale hotels geen probleem alleen ..... is dit geen normaal hotel. Dit is een doolhof. Het doolhof bevat 181 kamers en is verdeeld in twee etages, de tweede etage is pas te bereiken als de eerste geheel doorlopen is. Per kamer wordt een plattegrond gegeven en de mogelijke uitgangen. Heel veel sterkte en denk om de nachtrust.

```
 1 rem -------------------------------
   --
 2 rem --******* piccolo ************
   --
 3 rem -- door : peter boersma  1987
   --
 4 rem -- voorstekamp 62 stadskanaal
   --
 5 rem --  tel.:(05990-) 20402
   --
 6 rem --social software services (c)
   --
 7 rem -------------------------------
   --
 8 rem -- voor : commodore info
   --
 9 rem -------------------------------
   --
10 trap 30
20 goto 40
30 scnclr:printerr$(er)"error[SPACE]i
   n"el:help:end
40 gosub 1000:rem titel+intro
50 scnclr:char 1,11,11,"even[SPACE]ge
   duld[SPACE]a.u.b."
60 dim k$(9,9)
70 for y=1to9:forx=1to9
```

```
 80 read k$(x,y)
 90 next x,y
100 data bl,b,bo,b,bo,bor,bl,bo,ro
110 data lr,lro,lrb,lr,lb,br,lo,bo,br
120 data l,bo,ro,lr,lr,lro,lb,bo,r
130 data lr,lb,ob,,o,br,lr,lb,ro
140 data lr,lo,obr,lr,lb,o,or,lr,lrb
150 data l,ob,ob,r,lr,obl,b,ro,lr
160 data lr,bl,bo,r,l,ob,o,ob,r
170 data lr,lo,obr,lr,lor,lob,rb,obl,r
180 data lro,lbo,ob,o,ob,ob,o,ob,or
190 rem ---- hoofdprogramma ----
200 x=1:y=9:rem startpositie
210 char1,11,11,"[23xSPACE]":print
220 print"[3xCRSR-DOWN][SPACE][CTRL 6]
    je[SPACE]stapt[SPACE]uit[SPACE]de[
    SPACE]lift.je[SPACE]hebt[SPACE]de"
230 print"[SPACE]post[SPACE]in[SPACE]j
    e[SPACE]hand.[CTRL 8]probeer[SPACE
    ]maar[SPACE]bij"
240 print"[SPACE]kamer[SPACE]nr.0[SPAC
    E](nul)[SPACE]te[SPACE]komen!!"
250 print"[CRSR-DOWN][SPACE]dat[SPACE]
    kun[SPACE]je[SPACE]doen[SPACE]door
    [SPACE]er[SPACE]heen[SPACE]te[SPAC
    E]"
260 print"[2xSPACE]wandelen[SPACE]in[S
    PACE]de[SPACE]richting[SPACE]die[S
    PACE]je[SPACE]intypt."
270 print"[CRSR-DOWN][SPACE]dat[SPACE]
    intypen[SPACE]gebeurt[SPACE]zo:"
280 print"je[SPACE]typt[SPACE]:'n'[SPA
    CE]voor[SPACE]noord[SPACE](boven)
290 print"[9xSPACE]'z'[SPACE]voor[SPAC
    E]zuid[SPACE](onder)
300 print"[9xSPACE]'o'[SPACE]voor[SPAC
    E]oost[SPACE](rechts)
310 print"[9xSPACE]'w'[SPACE]voor[SPAC
    E]west[SPACE](links)
320 print"[2xCRSR-DOWN][3xSPACE]B[CTRL
    9][SPACE]veel[SPACE]succes[SPACE]
    !![SPACE][CTRL 0]"
330 gosub 1490:color1,8,5:scnclr
340 rem -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    -+
350 scnclr:n=0:o=0:z=0:w=0:z$=""
360 if y<1 and e1=0 then goto 2010:rem
     gelukt
364 n=0:o=0:z=0:w=0:n$="":o$="":u$="":
    w$="":gaa=0
365 gosub 370:if gaa=1 then goto 360
366 goto 350
370 be=be+1:n$="":o$="":u$="":w$=""
380 z$="":for a=1 to len(k$(x,y))+1
390 if mid$(k$(x,y),a,1)="r" then z$=z
    $+chr$(13)+"[3xSPACE]oost":o=1:o$=
    "[SHIFT -]"
400 if mid$(k$(x,y),a,1)="l" then z$=z
    $+chr$(13)+"[3xSPACE]west":w=1:w$=
    "[SHIFT -]"
410 if mid$(k$(x,y),a,1)="o" then z$=z
    $+chr$(13)+"[3xSPACE]zuid":z=1:u$=
    "[SHIFT +][3xSHIFT *][SHIFT +]"
420 if mid$(k$(x,y),a,1)="b" then z$=z
    $+chr$(13)+"[3xSPACE]noord":n=1:n$
    ="[SHIFT +][3xSHIFT *][SHIFT +]"
430 next a
435 nr=x*y+int(x*y/2.5)+1
440 print"[2xCRSR-DOWN]******[SPACE]ka
    mer[SPACE]nr.";nr;"[SPACE]********
    ***"
445 if z$="" then print"[3xCRSR-DOWN][
    3xSPACE]je[SPACE]kunt[SPACE][CTRL
    9]overal[CTRL 0][SPACE]heen!":goto
     465
450 print"[3xCRSR-DOWN][3xSPACE]je[SPA
    CE]kunt[SPACE][CTRL 9]niet[CTRL 0
    ][SPACE]naar[SPACE]:"
460 printz$
465 if o$="" then o$="[SPACE]"
470 if w$="" then w$="[SPACE]"
471 if n$="" then n$="[SHIFT +][3xSPAC
    E][SHIFT +]"
472 if u$="" then u$="[SHIFT +][3xSPAC
    E][SHIFT +]"
480 print"[CRSR-DOWN][COM 6]situatie:[
    SPACE]";n$
490 printtab(10)w$;"[3xSPACE]";o$
500 printtab(10)w$;"[SPACE]EVD[SPACE]"
    ;o$
510 printtab(10)w$;"[3xSPACE]";o$
520 printtab(10)u$;"[CTRL 8]"
530 print"[2xCRSR-DOWN][4xSPACE]waar[S
    PACE]wil[SPACE]je[SPACE]heen[SPACE
    ]?[SPACE](n/o/z/w)"
540 geta$:ifa$<>"n"anda$<>"z"anda$<>"o
    "anda$<>"w"then 540
550 if a$="n" and n=1 then gosub 630:r
    eturn
560 if a$="z" and z=1 then gosub 630:r
    eturn
570 if a$="o" and o=1 then gosub 630:r
    eturn
580 if a$="w" and w=1 then gosub 630:r
    eturn
590 if a$="n" then y=y-1:n=0:return
600 if a$="o" then x=x+1:o=0:return
610 if a$="z" then y=y+1:z=0:return
620 if a$="w" then x=x-1:w=0:return
630 print"[SHIFT CLR][CRSR-DOWN][CTRL
    9][SPACE]dat[SPACE]gaat[SPACE]niet
    [SPACE]!!":be=be-1
640 gaa=1:n=0:z=0:o=0:w=0:z$="":return
650 rem -------- titelpagina ---------
1000 scnclr:color0,1:color1,8,4:color4,
     1
1010 printchr$(27)"n";:print"[SHIFT CLR
     ][7xCRSR-DOWN]";
1020 color1,8,7:printspc(13)"[COM A]I.U
     IUIUI[SHIFT -]UI@'87"
1030 color1,8,5:printspc(13)"[4xSHIFT -
     ][SPACE][SHIFT -][SPACE][5xSHIFT -
     ][SPACE]by[SPACE]"
1040 color1,8,4:printspc(13)"[COM Q]K[S
     HIFT -]JKJKJK[SHIFT -]JK[SPACE]sss
     "
1050 color1,8,3:printspc(12)"U[COM E][S
     HIFT *][COM E][6xSHIFT *][COM E][4
     xSHIFT *]I"
1060 color1,8,4:printspc(12)"[SHIFT -][
     14xSPACE][SHIFT -]"
1070 color1,8,5:printspc(12)"J[14xSHIFT
      *]K
1080 color1,2,7:printchr$(142)chr$(8)
1090 t$="----------[7xSPACE]hallo[2xSPA
     CE]en[2xSPACE]welkom[2xSPACE]bij[2
```

```
     xSPACE]'piccolo'"
1100 t$=t$+",het[SPACE]nieuwste[SPACE]p
     rodukt[SPACE]van[SPACE]'social[SPA
     CE]software[SPACE]services'[SPACE]
     (sss)"
1110 t$=t$+".[3xSPACE]in[SPACE]dit[SPAC
     E]doolhof-spel[SPACE]bent[SPACE]u[
     SPACE]een[SPACE]piccolo[SPACE]die[
     SPACE]van[SPACE]zijn/haar"
1120 t$=t$+"[SPACE]baas[SPACE]een[SPACE
     ]opdracht[SPACE]krijgt.[2xSPACE]dr
     uk[SPACE]op[SPACE]een[SPACE]toets[
     SPACE]om[SPACE]die[SPACE]opdracht"
1130 t$=t$+"[SPACE]te[SPACE]weten[SPACE
     ]te[SPACE]komen....----"
1140 get b$:if b$<>"" then goto 1200
1150 t$=mid$(t$,2)+left$(t$,1)
1160 char 1,13,11,left$(t$,14)
1170 forc=1to25:nextc:goto1140
1180 :
1190 rem ---- gegevens ----
1200 color1,8,7:char 1,13,11,"[SPACE]pi
     ccolo[SPACE]data[SPACE]"
1210 fora=1to1000:nexta:scnclr
1220 color1,8,7:printspc(13)"[COM A]I.U
     IUIUI[SHIFT -]UI"
1230 color1,8,5:printspc(13)"[4xSHIFT -
     ][SPACE][SHIFT -][SPACE][5xSHIFT -
     ]"
1240 color1,8,3:printspc(13)"[COM Q]K[S
     HIFT -]JKJKJK[SHIFT -]JK"
1250 color1,8,1:print"[13xSHIFT *][COM
     E][SHIFT *][COM E][6xSHIFT *][COM
     E][17xSHIFT *]"
1260 printchr$(27)"t";
1270 print"[4xCRSR-DOWN][4xSPACE][CTRL
     6]welkom[SPACE]bij[SPACE][CTRL 8]p
     iccolo[CTRL 6]"
1280 print"[2xCRSR-DOWN][SPACE]stelt[SP
     ACE]u[SPACE]zich[SPACE]eens[SPACE]
     voor:
1290 print"[CRSR-DOWN][2xSPACE]u[SPACE]
     bent[SPACE]net[SPACE]als[SPACE]pic
     colo[SPACE]aangenomen[SPACE]en"
1300 print"[2xSPACE]u[SPACE]krijgt[SPAC
     E]de[SPACE]opdracht[SPACE]om[SPACE
     ]de[SPACE]post[SPACE]van"
1310 print"[2xSPACE][CTRL 8]kamer[SPACE
     ]nr.0[SPACE](nul)[CTRL 6][SPACE]te
     [SPACE]bezorgen."
1320 print"[CRSR-DOWN][SPACE]nu[SPACE]i
     s[SPACE]dat[SPACE]in[SPACE]normale
     [SPACE]hotels[SPACE]niet[SPACE]zo[
     SPACE]"
1330 print"[2xSPACE]moeilijk,aleen....i
     n[SPACE]dit[SPACE]hotel[SPACE]ligt
     "
1340 print"[2xSPACE]dat[SPACE]iets[SPAC
     E]anders....."
1350 gosub 1490
1360 print"[CRSR-DOWN][SPACE]de[SPACE]a
     rchitect[SPACE]van[SPACE]dit[SPACE
     ]hotel[SPACE]was[SPACE]op"
1370 print"[2xSPACE]z'n[SPACE]minst[SPA
     CE]gezegd[SPACE]niet[SPACE]helemaa
     l[SPACE]bij"
1380 print"[2xSPACE]z'n[SPACE]positieve
     n.(u[SPACE]kent[SPACE]dat[SPACE]we
     l:"
1390 print"[2xSPACE]een[SPACE]avondje[S
     PACE]stevig[SPACE]doordrinken[SPAC
     E]en[SPACE]een"
1400 print"[2xSPACE]koppijn[SPACE]dat[S
     PACE]je[SPACE]de[SPACE]volgende[SP
     ACE]ochtend"
1410 print"[2xSPACE]hebt!).":print"[SPA
     CE]goed,het[SPACE]hotel;"
1420 print"[CRSR-DOWN][SPACE]1)[SPACE]e
     r[SPACE]is[SPACE]maar[SPACE]1[SPAC
     E]verdieping:de[SPACE]vierde."
1430 print"[SPACE]2)[SPACE]er[SPACE]zij
     n[SPACE]kamernummers[SPACE]die[SPA
     CE]vaker[SPACE]dan"
1440 print"[4xSPACE]1[SPACE]keer[SPACE]
     voorkomen."
1450 print"[SPACE]3)[SPACE]sommige[SPAC
     E]kamernummers[SPACE]zijn[SPACE]er
     [SPACE]niet."
1460 print"[SPACE]4)[SPACE]de[SPACE]vie
     rde[SPACE]verdieping[SPACE]is[SPAC
     E]een[SPACE]doolhof!"
1470 print"[CRSR-DOWN][SPACE]oftewel:[S
     PACE][CTRL 8]veel[SPACE]succes[C
     TRL 6]"
1480 gosub 1490:return
1490 print"[2xCRSR-DOWN][8xSPACE][CTRL
     2]druk[SPACE]op[SPACE][[CTRL 9]ret
     urn[CTRL 0]][CTRL 6]"
1500 geta$:ifa$<>chr$(13)then1500:elser
     eturn
2000 rem ------ gelukt ! --------
2010 scnclr:ei=1
2020 print"[CTRL 8][4xCRSR-DOWN][2xSPAC
     E]het[SPACE]is[SPACE]je[SPACE]gelu
     kt!!"
2030 print"[2xSPACE]je[SPACE]staat[SPAC
     E]voor[SPACE]de[SPACE]trap[SPACE]n
     aar[SPACE]de[SPACE]"
2040 print"[SPACE]vijfde[SPACE]verdiepi
     ng!!!!
2050 fora=1to2000:nexta
2060 print"[CRSR-DOWN][SPACE]wacht[SPAC
     E]even[SPACE].......[CTRL 8]vijfde
     [SPACE]verdieping???[CTRL 6]"
2070 print"[3xSPACE]is[SPACE]er[SPACE]d
     an[SPACE]een[SPACE]vijfde[SPACE]ve
     rdieping???"
2080 print"[SPACE]ja,hoor;[SPACE][CTRL
     9]er[SPACE]is[SPACE]nog[SPACE]een[
     SPACE]vijfde"
2090 print"[3xSPACE][CTRL 9]verdieping!
     !![CTRL 0]"
2100 print"[SPACE]en[SPACE]je[SPACE]kun
     t[SPACE]er[SPACE]wel[SPACE]op[SPAC
     E]rekenen[SPACE]dat[SPACE]die"
2110 print"[2xSPACE]nog[SPACE]lastiger[
     SPACE]is!!"
2120 print"[CRSR-DOWN][SPACE]dat[SPACE]
     wordt[SPACE]dus[SPACE]weer[SPACE]z
     oeken!!"
2130 gosub 1490:scnclr
2140 for a=1 to 15
2150 for v=8 to 3 step -2
2160 volv-a/5:sound3,550,1:sound3,565,1
     :sound3,580,1:sound3,595,1:sound3,
     610,1
2170 next v,a:scnclr
```

```
2180 char 1,11,11,"[CTRL 7]even[SPACE]geduld[SPACE]a.u.b."
2190 for y=0 to 9
2200 for x=0 to 9
2210 read k$(x,y):next x:next y
2220 data blo,br,bl,bo,br,bl,bo,b,bo,or
2230 data lbr,l,ro,lrb,lro,lro,lb,ro,lb,br
2240 data l,o,b,o,b,br,lo,b,or,lr
2250 data lro,lb,ro,lrb,lro,l,ob,r,lb,or
2260 data lrb,lr,lob,o,rb,lo,rob,lr,lo,obr
2270 data l,-,ob,obr,lo,b,ob,o,ob,br
2280 data lr,lo,ob,br,lob,or,lrb,lb,ob,or
2290 data l,b,b,r,lrb,lb,-,o,ob,br
2300 data lr,lro,lr,lo,r,lr,lr,lb,bor,lr
2310 data lo,ob,o,obr,ol,or,ol,or,olb,or
2320 x=0:y=9:color1,8,4
2330 scnclr:if y<0 then goto 4000:rem einde 2
2340 n=0:z=0:o=0:w=0:n$="":o$="":u$="":w$="":gaa=0
2350 gosub 370:if gaa=1 then goto 2340
2360 goto 2330
4000 print"[SHIFT CLR][6xCRSR-DOWN]";
4010 color1,8,7:printspc(10)"[COM A][SHIFT *]IU[SHIFT *]IU[SHIFT *]IUI[SPACE]UI[3xSPACE]U[SHIFT *]I"
4020 color1,8,6:printspc(10)"[SHIFT -][SPACE][2xSHIFT -][SPACE][2xSHIFT -][SPACE][3xSHIFT -][SPACE][2xSHIFT -][SPACE]nr[SHIFT -][SPACE][SHIFT -]"
4030 color1,8,5:printspc(10)"[COM Q][COM R]K[SHIFT -][SPACE][2xSHIFT -][SPACE][2xSHIFT -]J[SHIFT *]K[SHIFT -][3xSPACE][SHIFT -][SPACE][SHIFT -]"
4040 color1,8,4:printspc(10)"[SHIFT -]JIJ[SHIFT *]KJ[SHIFT *]K[SHIFT -][3xSPACE][SHIFT -][3xSPACE]J[SHIFT *]K"
4050 color1,8,3:printspc(10)"[COM E][SHIFT *][COM E][6xSHIFT *][COM E][3xSHIFT *][COM E][6xSHIFT *]"
4060 print"[3xCRSR-DOWN]";:gosub 1490
4070 run

** EINDE LISTING piccalo c16
```

REGEL 1	47	REGEL 290	174	REGEL 580	187	REGEL 1290	168	REGEL 2150	30
REGEL 2	106	REGEL 300	253	REGEL 590	77	REGEL 1300	43	REGEL 2160	32
REGEL 3	19	REGEL 310	187	REGEL 600	76	REGEL 1310	106	REGEL 2170	103
REGEL 4	45	REGEL 320	221	REGEL 610	100	REGEL 1320	83	REGEL 2180	1
REGEL 5	13	REGEL 330	148	REGEL 620	93	REGEL 1330	158	REGEL 2190	153
REGEL 6	97	REGEL 340	15	REGEL 630	209	REGEL 1340	142	REGEL 2200	152
REGEL 7	47	REGEL 350	84	REGEL 640	224	REGEL 1350	91	REGEL 2210	77
REGEL 8	148	REGEL 360	87	REGEL 650	190	REGEL 1360	218	REGEL 2220	198
REGEL 9	47	REGEL 364	2	REGEL 1000	94	REGEL 1370	43	REGEL 2230	231
REGEL 10	58	REGEL 365	97	REGEL 1010	67	REGEL 1380	138	REGEL 2240	45
REGEL 20	237	REGEL 366	33	REGEL 1020	251	REGEL 1390	56	REGEL 2250	86
REGEL 30	74	REGEL 370	52	REGEL 1030	100	REGEL 1400	66	REGEL 2260	231
REGEL 40	80	REGEL 380	131	REGEL 1040	163	REGEL 1410	114	REGEL 2270	222
REGEL 50	4	REGEL 390	239	REGEL 1050	21	REGEL 1420	106	REGEL 2280	139
REGEL 60	228	REGEL 400	247	REGEL 1060	223	REGEL 1430	191	REGEL 2290	64
REGEL 70	109	REGEL 410	13	REGEL 1070	25	REGEL 1440	35	REGEL 2300	112
REGEL 80	36	REGEL 420	51	REGEL 1080	171	REGEL 1450	17	REGEL 2310	99
REGEL 90	95	REGEL 430	195	REGEL 1090	46	REGEL 1460	15	REGEL 2320	206
REGEL 100	186	REGEL 435	217	REGEL 1100	65	REGEL 1470	242	REGEL 2330	61
REGEL 110	206	REGEL 440	97	REGEL 1110	62	REGEL 1480	35	REGEL 2340	2
REGEL 120	251	REGEL 445	151	REGEL 1120	204	REGEL 1490	51	REGEL 2350	145
REGEL 130	80	REGEL 450	23	REGEL 1130	43	REGEL 1500	26	REGEL 2360	81
REGEL 140	153	REGEL 460	23	REGEL 1140	205	REGEL 2000	242	REGEL 4000	17
REGEL 150	159	REGEL 465	4	REGEL 1150	179	REGEL 2010	123	REGEL 4010	181
REGEL 160	1	REGEL 470	20	REGEL 1160	125	REGEL 2020	217	REGEL 4020	33
REGEL 170	42	REGEL 471	184	REGEL 1170	58	REGEL 2030	154	REGEL 4030	137
REGEL 180	48	REGEL 472	198	REGEL 1180	58	REGEL 2040	228	REGEL 4040	76
REGEL 190	13	REGEL 480	214	REGEL 1190	75	REGEL 2050	8	REGEL 4050	230
REGEL 200	45	REGEL 490	110	REGEL 1200	219	REGEL 2060	243	REGEL 4060	224
REGEL 210	112	REGEL 500	74	REGEL 1210	41	REGEL 2070	29	REGEL 4070	138
REGEL 220	92	REGEL 510	110	REGEL 1220	37	REGEL 2080	192		
REGEL 230	53	REGEL 520	92	REGEL 1230	201	REGEL 2090	209		
REGEL 240	224	REGEL 530	48	REGEL 1240	169	REGEL 2100	18		
REGEL 250	238	REGEL 540	154	REGEL 1250	195	REGEL 2110	250		
REGEL 260	26	REGEL 550	169	REGEL 1260	237	REGEL 2120	132		
REGEL 270	223	REGEL 560	193	REGEL 1270	168	REGEL 2130	125		
REGEL 280	4	REGEL 570	171	REGEL 1280	151	REGEL 2140	175		

CITY-DESTROYER

Peter Boersma maakt van het totale geheugen van de c-16 gebruik bij het programma CITY-DESTROYER. Daarom is het zaak de regelnummering niet te veranderen. Het spel maakt gebruik van het grafische scherm. De bedoeling van het spel is het afwijkend gekleurd flatgebouw te vernietigen. Dit kan gebeuren met het kanon links op het scherm. Er moet rekening worden gehouden met: de wind, de vuursnelheid en het aantal graden waaronder gevuurd moet worden.

```
1 sc=5
2 scnclr:color0,7,2:color1,2:color4,
  1
3 graphic2,1:i=int(rnd(1)*20-9):i$=s
  tr$(i)
4 color 1,2:vol 8
5 g=20:gosub 6:goto 9
6 box1,40,110,44,140,g
7 char 1,10,0,"[4xSPACE]":sc$=str$(s
  c)
8 char 1,10,0,sc$:return
9 char1,37,0,i$:char1,32,0,"wind:"
10 char 1,0,0,"bull.left:"
11 draw 1,42,125 to 30,145
12 draw 1,42,125 to 47,145
13 draw 1,25,145 to 50,145
14 draw 1,25,145 to 0,160
15 draw 1,50,145 to 200,160
16 draw 1,0,160 to 200,160
17 color 1,6,5:paint1,3,159:color 1,2
18 draw 1,195,159 to 320,159
19 a=205:b=6:q=0:gosub 24
20 a=220:b=12:q=0:gosub 24
21 a=211:b=13:q=41:gosub 24
22 a=300:b=8:q=35:gosub 24
23 a=306:b=14:q=0:gosub 24:goto 29
24 ya=rnd(1)*40+100-q
25 color 1,1:box 1,a,ya,a+b,160
26 paint1,a+b-5,ya+2:paint1,a+b-4,ya+
   2
27 forc=atoa+bstep3:fore=ya+1to159ste
   p3
28 draw0,c,e:nexte:nextc:color1,2:ret
   urn
29 u=rnd(1)*45+240:w=rnd(1)*50+75
30 color 1,16,5:draw 1,u,w to u+10,w
31 draw 1,u,w to u,160
32 draw 1,u+10,w to u+10,160
33 draw 1,u,160 to u+10,160
34 paint 1,u+5,w+5
35 for d=u+2 to u+10 step 3
36 for f=w+1 to 159 step 3
37 draw 0,d,f:next f:next d
38 paint 1,u+5,w+5
39 char 1,int(u/8)-1,5,"city"
40 color 1,2:forl=1to250:next
41 sc=sc-1:ifsc<0thenf$="failed":goto
   69
42 print"[HOME]";:forn=1to20:print:ne
   xt
43 input"vuurkracht(1-10)";v:v=11-v
44 input"graden(1-90)";g
45 if v<1 or v>10 then goto 42
46 if g<1 or g>90 then goto 42
47 forn=0to90:box0,40,110,44,140,n:ne
   xt
48 gosub6:forl=1to300:nextl:sound3,1,
   9
49 x=42:y=125:v=-v:t=0
50 draw 0,x,y:draw 0,x,y-1
51 x=x+g/14+i/9:y=y+v/2
52 if t>0 then goto 54
53 if y<-20then v=-v:t=t+1
54 draw 1,x,y:draw 1,x,y-1
55 if x>u-1 and x<u+10 then goto 59
56 if y>160 then goto 41
57 if x>325 or x<0 then goto 41
58 goto 50
59 if y>w-2 and y<w+2 then goto 61
60 goto 50
61 vol8:draw0,x,y:draw0,x,y-1:sound1,
   1,5:sound3,9,20
62 vol 6:sound 3,10,20:color 1,9,5
63 for l=1 to 300:next l
64 for r=w to 157 step 2
65 draw 0,u,r to u+10,r:sound 3,1,2
66 draw 0,u,r+1 to u+10,r+1:next r
67 for l=1 to 500:next l:color 1,2
68 f$="completed"
69 graphic 0,1:forn=1to9:print:next
70 color0,1:print tab(12)"mission[SPA
   CE]";f$
71 for l=1 to 1500:next l:sc=sc+2
72 if f$="failed" then goto 1:else go
   to2
```

REGEL 1	125	REGEL 37	249
REGEL 2	92	REGEL 38	210
REGEL 3	142	REGEL 39	99
REGEL 4	195	REGEL 40	87
REGEL 5	84	REGEL 41	214
REGEL 6	40	REGEL 42	172
REGEL 7	187	REGEL 43	30
REGEL 8	168	REGEL 44	20
REGEL 9	176	REGEL 45	115
REGEL 10	251	REGEL 46	93
REGEL 11	57	REGEL 47	226
REGEL 12	65	REGEL 48	245
REGEL 13	62	REGEL 49	0
REGEL 14	6	REGEL 50	82
REGEL 15	102	REGEL 51	187
REGEL 16	46	REGEL 52	89
REGEL 17	255	REGEL 53	195
REGEL 18	176	REGEL 54	84
REGEL 19	136	REGEL 55	125
REGEL 20	178	REGEL 56	193
REGEL 21	232	REGEL 57	174
REGEL 22	190	REGEL 58	238
REGEL 23	231	REGEL 59	78
REGEL 24	208	REGEL 60	238
REGEL 25	16	REGEL 61	220
REGEL 26	15	REGEL 62	139
REGEL 27	170	REGEL 63	239
REGEL 28	109	REGEL 64	248
REGEL 29	64	REGEL 65	152
REGEL 30	231	REGEL 66	90
REGEL 31	214	REGEL 67	161
REGEL 32	236	REGEL 68	253
REGEL 33	33	REGEL 69	195
REGEL 34	210	REGEL 70	129
REGEL 35	136	REGEL 71	22
REGEL 36	202	REGEL 72	187

Vragen van gebruikers

Er blijkt in toenemende mate een grote behoefte te bestaan aan de mogelijkheid om vragen te stellen over allerlei Commodore-onderwerpen. Ook deze keer gaat Jan Bodzinga weer in op een aantal problemen. Uw vragen - maar ook uw tips en handigheidjes - zijn van harte welkom.

Conversie

De heer Benningshof uit Den Haag komt met een vraag die betrekking heeft op de mogelijkheden van dataconversie van C-64 gegevens naar andere systemen. Hij vraagt zich af, wat hiervan de mogelijkheden zijn.

Dit is zeker niet de enige brief die we over het fenomeen data-conversie hebben ontvangen. Het lijkt erop, dat nogal wat serieuze computeraars hun C-64 of andere (kleine) huiscomputer in de kast willen zetten en er een grotere, meer professionele machine voor in de plaats willen aanschaffen. Eén van de problemen, die daarbij om de hoek komt kijken, is de vraag of het mogelijk is bestaande C-64 bestanden over te zetten naar met name IBM-PC en compatiblele formaten. Zo ook deze briefschrijver.

Het antwoord op deze vraag is simpel, maar heeft wel de nodige complicaties. In principe kunnen alle bestanden zonder te veel moeite, via een interface, van de C-64/C-128 worden overgezet naar een PC-diskformaat. Voor Amiga-types is het helemaal geen probleem, mits de MS-DOS emulator beschikbaar is. In het ondehavige geval gaat het om een serie C-64 files, waarbij helaas geen gegevens zijn verstrekt over de aard van de data.
Om dergelijke bestanden -in hun geheel- te kunnen overzetten kan men gebruik maken van een RS-232 interface, dat aan de userpoort van de C-64 moet worden aangesloten. Dit interface (b.v. de 1110A van Commodore) kan met een standaard seriële kabel aan een serie-poort van de IBM of compatible computer worden gekoppeld, waarna met standaard RS232 protocols de data vanaf de C-64 disk of cassette kan worden overgeseind naar de PC. De maximale Baudrate die de C-64 daarbij kan verwerken ligt ongeveer bij 1600 Baud, ofwel maximaal zo'n 200 tekens per seconde. Het overzetten van grote bestanden gaat daardoor wel even duren, maar het lukt.
Wat er vervolgens nog moet gebeuren met deze bestanden is het vertalen van de afwijkende Commodore ASCII naar standaard ASCII-code, zoals deze bij de PC wordt toegepast. Daarna moet er natuurlijk ook nog een PC programma worden gevonden, wat met deze data kan werken. Voor geformateerde records met een vaste velden recordlengte is dit na vertaling geen probleem. Deze kunnen zonder meer door een pakket als dBase II of III(plus), VP-INFO etc. worden ingelezen als tekstbestand (.SDF). In andere gevallen zal de data eerst op de C-64 moeten worden voorbewerkt.
Kortom, alle gegevens kunnen probleemloos worden overgestuurd naar een PC, maar men moet daarbij wel beschikken over de juiste interface, die helaas wat moeilijk te verkrijgen valt. Bob Munniksma heeft voor Commodore Info ooit een schema voor een dergelijk interface ontwikkeld. (CI nov. 1986) Ook kunnen wij gegadigden helpen aan een paar adressen die commercieel bezig zijn met het overzetten van data. De kosten daarvan liggen gemiddeld rond de 100 tot 200 gulden per volgeschreven C-64 diskette.

Het converteren van C-64 programma's naar PC-formaat heeft geen enkele zin, omdat deze software niet werkt op een PC. Voor Basic is dit een beetje anders, maar na het overseinen van de ASCII-listing moet er aan de Basic meestal ook nog wel het nodige worden aangepast, voor het programma goed werkt op een PC.

Userpoort

Het werken met de Userpoort en het aansluiten van zelf ontwikkelde of gebouwde devices aan deze poort kan zorgen voor onverwachte problemen.

Zo heeft in het belgische Koekelare filip Lauwyck problemen na het vergeefs aansluiten van een zelfgebouwde Eprommer. De C-64 is daarna vreemde kuren gaan vertonen.
De structuur en aansluitingen van de User-poort zijn nogal gevoelig voor het fout aansluiten van interfaces. Doordat hierbij meestal wat ondoordacht te werk wordt gegaan komt het maar al te vaak voor, dat één of meer inkorrekte aansluitingen de CIA chips voor een gedeelte hebben vernield. Dit feit zal de directe werking van de C-64 niet schaden, maar pas aan het licht komen, als er I/O wordt gepleegd die ook via één van deze 6526 Complex Interface Adapters loopt.
Zo schrijft Filip bijvoorbeeld over een uitgangsspanning van 20 Volts, waarbij de CIA slechts maximaal +7V verdragen kan. Een fout zit daarbij in een klein hoekje, en in dit geval zal er ongetwijfeld zoiets aan de hand zijn. Zeker bij opdrachten die te maken hebben met kopiëren en moven van tekst krijgt men dan problemen met de C-64. Ons advies is daarom ook, de C-64 maar snel te laten testen, met het accent op de CIA's, en in het vervolg voorzichtiger te werk te gaan met de Userpoort.

Verslaving

Het computerspel **Fist il** zorgt in Rotterdam voor heel wat slapeloze nachten. Eelko Wolthof wil niets meer, behalve achter z'n computer dit spel spelen. Hij heeft daarbij helaas het probleem, dat ie maar niet langs de rotsblokken in de grotten kan komen, terwijl op andere momenten hij gewoon over ravijnen heen huppelt en de berg op komt zonder trap.

Voor mij eigenlijk allemaal onbekende fenomenen, maar ongetwijfeld zitten er onder de lezers nog wel de nodige andere verslaafden, die Eelko wel willen helpen, hoewel het feit dat Eelko met dit spel dingen kan doen, die in fewite niet horen bij de spelregels, kan het ook wel zijn, dat de Fist II kopie van Eelko niet helemaal goed werkt.
Wie wil hem helpen ?
Zijn adres is: *Claes de Vrieselaan 31b, 3021 JB Rotterdam*

Softwarebank

Vanuit Amsterdam een vraag over de legaliteit van software die vanaf bulletin-boards en databanken kan worden betrokken. K.W.T. van Hoogstraten stuurt een screendump op van één van 's lands databanken, die onder fraaie namen commerciële software te koop aanbied tegen zeer gereduceerde prijzen.

In Nederland nemen we het helaas niet erg nauw met auteursrechten en andere vormen van bescherming tegen illegale handelingen met andermans bezit. Ook de juridische gevolgen van overtredingen zijn weinig angstaanjagend. Daarom wordt er hier te lande nogal wat gerommeld met originele software, die gekopieerd en wel weer als nieuw aan tweede gebruikers wordt aangeboden. Zo komt het nogal eens voor, dat iemand een 'echt' pakket kopt voor een normale prijs. Dit programma wordt keurig op backup-schijven gezet, een voorziening die de fabrikant om begrijpelijke redenene toelaat, de manuals worden netjes gekopieerd en het originele pakket gaat (meestal zonder registratiekaart) weer in de doos en wordt voor een schappelijke prijs voor de tweede, derde of X-ste keer aan de man gebracht. Niet erg wettig allemaal, maar wat kun je er anders tegen doen, dan dit soort aanbiedingen te negeren.
Anders wordt het echter wanneer andermans eigendommen te koop worden gezet op databanken en bulletinboards, waar iedereen die in het bezit is van een modem tegen betaling aan de sysop, maar software vanaf kan halen. Dit isa volstrekt illegaal, en we zullen er dan ook stappen tegen nemen. Wat wel kan, is dat iemand met een softwarebedrijf oude partijen originele software opkoopt en alleen de aanbiedingen via de databank bekend maakt. Een soort MODEM-orderbedrijfje dus. Koopt men een dergelijk programma, bekijk het dan nauwkeurig en ga na wat er met de registratiekaart is gebeurd. Lijkt alles in orde, dan kun je het programma zonder probleem als legaal bezit beschouwen.

Printproblemen

P.Gruijters uit s'-Gravenhage komt met een serie trieste voorbeelden van alles wat z'n computer wel kan, maar z'n Star printer niet erg goed begrijpt.

Als eerste advies moet ik helaas kwijt, dat alle software, die specifiek ontwikkeld wordt voor de C-64/C-128 en eventueel andere Commodore huiscomputers, daarbij ook vaak uitgaat van een specifieke Commodore printer. Als prima compatible printers voor deze gekke Commodore-aansturing zijn er helaas op de markt geen of bijzonder weinig machines te vinden. De keuze voor een 100% goede configuratie behelst dus zonder meer een Commodore-printer. Koopt men, ondanks dit, toch een andere afdrukker, dan kan het niet anders of er komen bij diverse soorten software problemen om de hoek kijken. Ik ben het direct met iedereen eens, die zegt dat er veel betere, goedkopere, sneller printers te koop zijn, dan de Commodores. 't Is waar, maar de software is nu eenmaal in de eerste plaats voor een echte Commodore configuratie gemaakt en als het daarnaast ook nog werkt op andere printers inclusief (meestal weer een ander) interface, kun je alleen van geluk spreken. Commodore is nu eenmaal altijd wat eigenwijs in dit soort dingen.
Bepaalde problemen zijn echter wel het hoofd te bieden. Zo zullen we strak wat printeraansturing voor de Star-Gemini's onder de loep nemen. Ook wil het nog wel eens gebeuren, bij grafische programma's als Newsroom en Printmaster, dat er door een foute seriële overdracht op bijvoorbeeld 7 bits woorden gewerkt wordt in plaats van 8 bits, of omgekeerd. In dat geval krijgt men wel iets op de printer, maar dit bestaat dan uit allemaal vreemde printtekentjes, als omgekeerde vraagtekens, hoekjes, streepjes en dergelijke.
Deze problemen kunnen na wat geëxperimenteer en geduld wel door de gebruiker worden opgelost. Andere problemen en weigeringen van de printer kunnen alleen worden verholpen door de aankoop van een echte (grafische) Commodore printer. Daarmee verzekert men zich van een 100% compatible configuratie.
Overigens wil ook een ander, beter en duurder interface tussen vreemde printer en 128/64 ook nog wel eens helpen.

> Verreweg het grootste gedeelte van de vragen voor deze rubriek heeft betrekking op het werken met printers. Vooral de niet-Commodore type printers worden nog al eens gekocht als aanvulling van de configuratie, waarbij een interface dan moet zorgen voor een juiste aansturing vanuit de computer. De grote hoeveelheid brieven over problemen hiermee, bewijst wel, dat het lang niet altijd zo goed gaat als de printerverkopers willen doen geloven.

Spelletjes kopiëren

Het blijkt voor M.Teutelink uit Enschede niet mogelijk om zomaar spelletjes van cassette naar disk te kopiëren. Vooral software die groter is dan 151 blokken gaat er bij hem (en vele anderen) niet door. Ook kopieerprogramma's laten het daarbij afweten.

> Het is natuurlijk niet legaal om andermans software straffeloos te kopiëren. Alleen voor eigen gebruik, zoals bijvoord het maken van een backup, is dit min of meer toegestaan. Vaak is dit soort software daarom beveiligd tegen het maken van duplicaten. En terecht, vinden wij. Wil je desondanks toch een kopie maken van dit type software, dan zul je daar zelf een oplossing voor moeten vinden.

Waar we wel mee kunnen helpen is een listing voor het overzetten van legale (eigen) software van cassette naar disk:

```
10   REM KOPIEREN CASSETTE
     naar DISK
20   REM Commodore INFO jb 0987
100  REM START
110  F1$ = "PRG1,P,R" : REM
     CASSETTE
120  F2$ = "PRG1,P,W" : REM DISK
200  REM OPEN CASSETTE PRG
210  OPEN 1,1,0,F1$
220  OPEN 2,8,2,F2$
260  EF = 0
300  GET#1,A$
310  IF A$="" THEN A$=CHR$(0)
320  IF ST <> 0 THEN EF = 1:REM
     STATUS
400  PRINT#2,A$;
410  IF EF = 0 THEN 300
500  CLOSE 1: CLOSE 2
600 END
```

Dit eenvoudige programma laat de manier zien, waarop een basic programma vanaf cassette byte voor byte naar disk wordt geschreven, zonder eerst in het computer RAM te hoeven worden geladen.

Op deze manier kan ook een erg lange listing worden over gezet. Voor het kopiëren van commerciële software werkt dit programma echter niet. Ook bepaalde types cassette-record hebben een bug in de headerfile, waardoor na het lezen van de eerste 256 bytes het programma jammer genoeg stopt.
Alle andere types werken echter prima met een dergelijk programma.

Modems

Over modems krijgen we ook nogal wat vragen. Zo komt van M.J. de Vaan-Schetselaar de vraag wat nu precies het verschil is tussen een PTT-modem en een Teltron. Technische specificaties zeggen de gebruiker in spé meestal weinig, de prijs des te meer.

Een modem is een apparaat, wat door aansluiting op zowel computer als telefoonlijn kan zorgen voor een standaardcommunicatie tussen twee computers of de eigen computer en een databank c.q. bulletinboard. Het aardige hierbij is, dat de standaard communicatie ervoor kan zorgen, dat het niet meer uitmaakt, of de computers aan beide kanten van de lijn van hetzelfde type zijn. De overgeseinde gegevens worden namelijk aan beide kanten 'vertaald' naar één van de standaard protocols in piepjes en biepjes met een bepaalde betekenis. De modems aan beide zijden van de lijn MOduleren en DEModuleren deze serie geluidjes netjes naar gegevens die weer door de computer gebruikt kunnen worden.

Het enige waarbij voor de aankoop van een modem op moet worden gelet, is de capaciteit van zo'n device en de aansluitmogelijkheden voor de eigen computer. Een Commodore huiscomputer heeft meestal niet de beschikking over een standaard RS232 uitgang. Dat is nu juist de poort waarmee een gewoon modem het liefst werkt. Soms, zoals bij veel PC modems zit deze poort reeds in het modem gebouwd, waardoor het apparaat zelfs in de computer is te bouwen. Bij Commodore moet men een keuze maken tussen specifiek modems voor de Commodore, die de vreemde C-aansluiting zelf vertalen in standaard RS232, maar deze modems zijn op geen enkel ander apparaat te gebruiken, of men moet een RS232 interface op de Commodore-machine kopen, en daaraan een standaardmodem koppelen. De laatste oplossing is ons inziens de beste, zeker, omdat met een dergelijk modem ook andere computers kunnen worden gebruikt. En de ontwikkeling gaat snel.
Verder bestaan en modems die werken met verschillende protocols. Deze protocols zijn niet meer dan een serie afspraken om het modemverkeer te standaardiseren. In Nederland werken we veel met het zo geheten CCITT-protocol, waarbij sprake kan zijn van verschillende zend/ontvang snelheden. Deze snelheden hebben alle een naam, waarbij V21 staat over een 300 Baud/sec Full Duplex overdracht. Dit is niet erg snel, maar werkt wel voor beide zijden even snel, ongeveer 300/8 = 35 tekens per seconde. Een modem van dit type hoeft niet meer te kosten dan 250 gulden. Het V22 protocol doet het veel beter, daarmee kunnen 1200 Baud/sec FD worden verstuurd en ontvangen. Een vier keer sneller apparaat dan de V22 devices dus. Wel moet er daarbij om worden gedacht, dat aan beide kanten van de lijn eenzelfde protocol wordt gehanteerd, anders begrijpen de modems elkaar niet. Vaak hebben de V22 modems ook de optie V21 ingebouwd, maar deze V22 modems zijn een heel stuk duurder, ongeveer 800 gulden. Als derde mogelijkheid voor huiscomputers bestaat er nog een V23 protocol, waarmee met 1200/75 Baud/sec kan worden gewerkt. Hier gaat het met grote snelheid één kant uit (1200) terwijl de informatie naar de andere kant met slechts 75 Baud/sec kan worden verstuurd. Viditel etcetera werken met dit protocol. Een dergelijk modem kost plusminus 400 - 500 gulden, terwijl dit, in combinatie met V21 zo'n 100 gulden meer wordt. Ook combinatie-modems van deze drie protocols zijn verkrijgbaar, maar kosten op dit moment nog steeds bijna 1000 gulden. Sneller dan 1200 Baud/sec kan ook, maar dan praten we al bijna over bijzonder professionele apparaten, die niet erg geschikt meer zijn voor de Commodore huiscomputers, omdat deze intern meestal niet meer dan 1600 Baud/sec. kunnen verwerken.

Grafisch printen

R.A. Koeroo in Groningen kan z'n Rambo-tekening niet mooi op papier krijgen met een General Electric TXP 1000. Het probleem daarbij is, dat de (nieuwe) printer tijdens het printen steeds een witte regelruimte openlaat, en dat geeft natuurlijk niet het gewenste effect voor de familie Koeroo.

Het door u geschetse probleem is niet onbekend en kan ook vrij eenvoudig worden verholpen. In de printer bevinden zich een aantal dip-schakelaars, waarmee de standaardinstelling van de printer te stellen is. Eén van deze schakelaars is bestemd voor de AUTO-LINEFEED. Deze kan zowel actief als passief worden gezet. Door het printerhandboek te raadplegen kunt u bekijken waar deze schakelaar zit, en hoe de instelling op dit moment is. Omdat bij het grafisch printen de software zelf zorgt voor een LINE-FEED wanneer dit nodig is, hoeft de printer dat niet ook nog eens te doen, zoals in dit geval wel gebeurt. Door deze schakelaar om te zetten, en de printer daarna uit en aan te schakelen, wordt een dergelijk probleem verholpen.

64/128 conversie

A.D.Verspuij vraagt naar tabellen en opzoekschema's die betrekking hebben op de interne verschillen tussen de C-128(D) en de oudere C-64 machines.

Om Basic listings op een C-64 te kunnen gebruiken, die voor de 128 zijn geschreven komt er echter nog wel wat meer kijken, dan alleen een paar conversietabellen voor PEEK en POKE. In de richting van C-64 naar C-128 gaat dit veel beter. Hoewel er daarbij altijd moet worden gerekend met een forse serie ombouwregels, kun je stellen, dat bijna alle C-64 software in 128-mode te gebruiken is. Helaas beschikken wij ook niet over een tabel, waarin alle C-128 adressen zijn verwerkt.

Wellicht is er onder de lezers iemand die dit allemaal al haarfijn heeft uitgezocht. We houden ons aanbevolen voor een dergelijke lijst en zullen die ook zeker voor algemeen gebruik willen publiceren.

Screendump

Eric van Gaal wil graag van bepaalde spelletjes een screendump maken. Hij vraagt hoe dat het beste te bereiken is.

Het maken van grafische screendumps, zeker van gekochte programma's is niet altijd even eenvoudig.
Hoewel we in het verleden wel diverse hulpmiddelen in de vorm van software voor dit soort dumps op papier hebben geplaatst, blijkt in de meeste gevallen een hardware oplossing beter te voldoen. Om grafische screendumps te kunnen maken kan het beste een cartridge worden gekocht, die deze actie voor de gebruiker kan uitvoeren. Er zijn diverse cartridges op de markt, die in meer of mindere mate in staat zijn van een grafisch schermbeeld afdrukken (in zwart/wit) op de printer te realiseren. De 'Final Cartridge II' van HP is bijvoorbeeld zo'n hulpmiddel waarmee onder andere aardige grafische schermdumps op de printer kunnen worden gemaakt. Ook zijn er natuurlijk andere cartridges die hier je hier een eind op weg

mee helpen. Alleen het afdrukken van sprites op papier wil nog wel eens wat problemen geven, omdat de sprites op een heel andere manier worden opgebouwd en op het scherm gezet, dan een standaard HI-res plaatje. Om deze dumps te maken, moet je natuurlijk wel beschikken over een printer met grafische mogelijkheden.

Printercodes

Uit Odijk komt de vraag van E.Klein, hoe hij zijn Star-Gemini printer goed kan configureren voor het printen van bepaalde afwijkende lettertypes en andere tekstattributen.

Het werken met een Gemini drukker voor ASCII teksten is in samenwerking kingkingmet de Commodore computers geen bezwaar. Alleen bepaalde tekstcodes wijken af van de standaard Commodore codes. Hieronder laten we de belangrijkste afwijkingen volgen:

SUPSCRIPT AAN	PRINT#4,CHR$(27);CHR$(83);CHR$(1)
SUBSCRIPT UIT	PRINT#4,CHR$(27);CHR$(84)
SUPERSCRIPT AAN	PRINT#4,CHR$(27);CHR$(83);CHR$(0)
SUPERSCRIPT UIT	PRINT#4,CHR$(27);CHR$(84)
ONDERSTREPEN AAN	PRINT#4,CHR$(27);CHR$(45);CHR$(1)
ONDERSTREPEN UIT	PRINT#4,CHR$(27);CHR$(45);CHR$(0)
VERGROTEN AAN	PRINT#4,CHR$(27);CHR$(69)
VERGROTEN UIT	PRINT#4,CHR$(27);CHR$(70)
DUBBEL AAN	PRINT#4,CHR$(27);CHR$(71)
DUBBEL UIT	PRINT#4,CHR$(27);CHR$(72)

Deze codes kunnen als programmaregel worden meegerund, mits vooraf met hulp van de opdracht OPEN 4,4 een kanaal naar de printer is geopend. Ook andere codes uit het printerboek kunnen op deze manier worden geïmplementeerd in een listing. Zelfs voor het werken met standaardprogramma's kunnen deze codes vooraf in een kleine eigen Basic-listing soms wonderen verrichten. Dit kan helaas alleen, als de software waar het om gaat de printer niet opnieuw RESET, want dan zijn alle bestaande codes weer uit het printergeheugen verwijdert.

Kleine annonces zonder commercieel oogmerk en zonder vraag en aanbod van illegale software worden gratis geplaatst. Stuur een kaartje naar Postbus 112, 1260 AC Blaricum



"Goto!"

Het NOS-radioprogramma Hobbyscoop is al jaren bezig om met de luisteraars te communiceren via de huiscomputer. Een van de middelen die men aanwendde was de invoering van het 'Esperanto voor de computer', de NOS-Basicode. Nu gaat men het versturen van programma's ook op een ander manier uitvoeren, namelijk via een soort draadloos digitaal net.

NOS digitaal de lucht in.

De radio gebruiken om computerprogramma's uit te zenden, dat is in de loop der jaren bijna een soort zendingswerk voor Hans Janssen en zijn medewerkers. De Hobbyscoop/Basicode activiteiten lopen al vanaf 1979.



Hans Jansen ontvangt een PC voor de digitale plannen

Sindsdien zijn er al heel wat verschrikkelijke cirkelzaag geluiden uitgezonden over de Hilversumse zenders. Vele duizenden luisteraars kregen zo programmatuur voor de huiscomputer gratis huis-aan-huis bezorgd.

In gesprek

Omdat mensen wel eens een uitzending missen (er zijn twee uitzendingen per week: op zondag en woensdag) heeft het programma ook een computer- databank ingesteld. Met een huiscomputer en een modem kunnen de luisteraars alle gegevens over de uitzending opvragen. Ook programmatuur tuurtuurdie men gemist mocht hebben, is in de Hilversumse databank beschikbaar.
Deze gegevensbank blijkt een groot succes bij de luisteraars: hij is bijna voortdurend in gesprek.

Dan maar de lucht in

Als echte radiomensen keek men om zich heen en wil nu de mogelijkheden van de ether nog verder exploreren. Een volgend poject van het radioprogramma zal dan ook worden uitgevoerd in samenwerking met tienduizenden nederlandse zendamateurs. Met de verenigingen op dit gebied is Hobbyscoop nu bezig een volledig draadloos digitaal net op te bouwen in de ether. Het gaat om het zogeheten Packet-radio project. De centrale zend/ontvanger hiervoor komt in Hilversum op het dak van het NOS-hoofdgebouw te staan.

Radio-6

Om in aanmerking te komen voor een vergunning van de PTT-Radio Controle Dienst was voor het Packet-radio project een geavanceerde computer nodig. Daar de financiële positie van het programma een aanschaf van een dergelijk apparaat niet toestond schenen de ontwikkelingen toch wat vertraagd te zullen worden. Een Nederlands computerbedrijf (G-2) vond de plannen van Hobbyscoop echter zo aantrekkelijk dat men spontaan besloot deze computer gratis beschikbaar te stellen aan het project. Door deze onverwachte ontwikkeling zal het nu mogelijk zijn om dit nieuwe station, dat in de wandelgangen al Radio-6 genoemd wordt, reeds volgend jaar te openen.

Hobbyscoop

Voor wie nog niet mocht weten, wanneer Hobbyscoop haar computerklanken via de normale radio verspreidt, dat is op woensdag op radio 1/2 FM om 19u en op zondag op radio 5 AM om 22.40 u. Tijdens die uitzendingen niet alleen Basicode programma's, maar ook nieuws en commentaren uit het computerwereldje.

De cursus machinetaal is in volle gang. Tjipke van der Land heeft gelukkig ondanks drukke werkzaamheden voldoende tijd gevonden om weer een aflevering van onze machinetaalcursus voor de 65xx familie te kunnen schrijven. In deze achtste aflevering gaan we verder met de basisbegrippen die nodig zijn om een professioneel machinetaal programma te kunnen schrijven. Tussen de bedrijven door maken we ook onze eerste echte programma's, die we op de computer kunnen runnen.

Cursus Machinetaal

Deel 8 : Hulpmidellen

In de vorige afleveringen hebben we allerlei basale onderwerpen behandeld, die nodig zijn om in machinetaal te kunnen programmeren. Zo kwamen onder meer het hexadecimale stelsel en begrippen als mnemonics en adresseermethoden aan de orde. We zijn nu op het punt aangeland, dat we deze kennis aan de praktijk kunnen toetsen.

We pakken de draad van deze cursus op bij het punt waar we in aflevering 7 zijn blijven steken. We zijn bezig met een eerste programma, wat we als voorbeeld gebruiken om je eigen naam op het scherm zichtbaar te maken. Eigenlijk kan dit op twee manieren, waarvan maar één de voorkeur geniet, uiteraard. Je kunt een naam rechtstreeks in het geheugen POKEn, maar je kunt ook via de 6510 microprocessor een naam of andere tekst op het scherm zetten.

Tekstvoorbeeld

We zullen dit met een voorbeeld wat duidelijker maken. Het rechtstreeks in het geheugen POKEn is eigenlijk meer Basic dan machinetaal, want, hoewel je met POKE opdrachten bezig bent het interne RAMgeheugen direct aan te spreken, pas je daarbij geen enkele **machinetaal instructie** toe, je plaatst alleen maar de naar ASCII overgezette letters rechtstreeks in het videogeheugen, dus je maakt in principe geen gebruik van de machinetaal instructieset van de processor.

Hierna volgt het voorbeeld om rechtstreeks in het videogeheugen te POKEn:

NAAM	**Decimalen**	**Hexadecimalen**
J	74	4A
A	65	41
N	78	4E
	32	20
K	75	4B
L	76	4C
A	65	41
A	65	41
S	83	53
S	83	53
E	69	45
N	78	4E

In Basic ziet het voorbeeld met bovenstaande tekst "JAN KLAASSEN" er zo uit:

```
5    PRINT "CLR" :
6    PRINT CHR$(14) :REM SCHAKEL
     NAAR ASCI NIVO LAAG
10   REM PROGRAMMA VOOR HET
     POKE VAN EEN NAAM IN
     HET VIDEOGEHEUGEN
20   S = 1400 : REM START ADRES
     VOOR AF TE DRUKKEN
     NAAM
30   READ A
40   IF A = -1 THEN END
50   POKE S,A
60   S = S + 1
70   GOTO 30
100  DATA
     64,65,78,32,75,76,65,65,83,83,6
     9,78,-1
```

In regel 100 vinden we de decimale getallen terug, waarmee de naam JAN KLAASSEN op het scherm wordt gepoked. Wil je dit vervangen door je eigen naam of een andere tekst, dan kun je de datagetallen voor de letters die je nodig hebt aflezen uit de ASCII-tabel die bij de vorige aflevering (C-INFO j4/nr5) op bladzij 12 is afgedrukt.

Tot zover het Basic POKEn van een tekst met decimale bytewaarden (ASCII). Het voorbeeld waar we straks mee te maken zullen krijgen is de 'echte' machinetaal, waar opdrachten rechtstreeks met behulp van de processor worden uitgevoerd.

In een programma, waar je uiteraard ook moet werken met tekst, krijg je op geen enkele wijze de kans om tekst, hoe dan ook, op de wijze zoals hierboven, met hulp van DATA en POKE, in het (video)geheugen te plaatsen, simpelweg omdat je dan in machinetaal bezig bent en niet in Basic. Dus hier moet je echt door middel van LDA instructies ergens tekst uit het geheugen halen en volgens de regels van het programma op de goede plaats in het videogeheugen neerzetten.

Om die reden gaan we in het volgende voorbeeld laten zien hoe je eerst nog op een simpele manier je naam op het scherm kunt laten schrijven. We kunnen dat doen door op de **immediate** manier de accumulator te la-

den met het het juiste ASCII getal, maar we kunnen ook eerst door middel van het POKE programma een naam ergens willekeurig in het gewone RAM-geheugen plaatsen in plaats van direct op het scherm, en daarna met een machinetaalprogramma de naam uit dezelfde adressen in het geheugen halen en daarna pas plaatsen in het videogeheugen.
Aangezien dit de manier is die het meest gaat voorkomen, gaan we dit voorbeeld op een hele eenvoudige manier uitvoeren waardoor je goed kunt begrijpen wat er gebeurt. De volgende keer gaan we deze veel te omslachtige manier nog een keer uitvoeren, alleen dan met behulp van een nieuwe instructie, waardoor het programma veel kleiner wordt, en daardoor ook sneller en gemakkelijker leesbaar.

```
10  REM poke programma voor
    machinetaal
20  S = 8192 : REM START ADRES
    NAAM
30  READ A
40  IF A = -1 THEN END
50  POKE S,A
60  S = S + 1
70  GOTO 30
100 DATA
    74,65,78,32,75,76,65,65,83,83,6
    9,78,-1
```

Het programma:

MNEMONIC	HEX	COMMENTAAR
LDA #$17	A9 17	;Getal laden voor asci nivo laag
STA $D018	8D 18 D0	;Parameter plaatsen
LDA $2000	AD 00 20	;Ophalen letter 1
STA $0578	8D 78 05	;Plaatsen van letter 1 in videogeh.
LDA $2001	AD 01 20	;Ophalen letter 2
STA $0579	8D 79 05	;Plaatsen van letter 2 in videogeh.
LDA $2002	AD 02 20	;Enz.
STA $057A	8D 7A 05	
LDA $2003	AD 03 20	;Ophalen van spatie
STA $057B	8D 7B 05	;Plaatsen van spatie in videogeh.
LDA $2004	AD 04 20	;Enz.
STA $057C	8D 7C 05	
LDA $2005	AD 05 20	
STA $057D	8D 7D 05	
LDA $2006	AD 06 20	
STA $057E	8D 7E 05	
LDA $2007	AD 07 20	
STA $057F	8D 7F 05	
LDA $2008	AD 08 20	
STA $0580	8D 80 05	
LDA $2009	AD 09 20	
STA $0581	8D 81 05	
LDA $200A	AD 0A 20	
STA $0582	8D 82 05	
LDA $200B	AD 0B 20	
STA $0583	8D 83 05	
NOP	EA	
BNE #$FC	D0 FC	;LOOP VIA NOP (NO OPERATION)

Alle Opcode's en Operands van bovenstaande listing achter elkaar in hexadecimaal:

```
A9 17 8D 18 D0 AD 00 20 8D 78 05
AD 01 20 8D 79 05 AD 02 20 8D 7A
05 AD 03 20 8D 7B 05 AD 04 20 8D
7C 05 AD 05 20 8D 7D 05 AD 06 20
8D 7E 05 AD 07 20 8D 7F 05 AD 08
20 8D 80 05 AD 09 20 8D 81 05 AD
0A 20 8D 82 05 AD 0B 20 8D 83 05
EA D0 FC
```

Alle Opcode's en Operands achter elkaar in decimaal (voor het poken):

```
169,23,141,24,208,173,0,32,141,120,
5,173,1,32,141,121,5,173,2,32
141,122,5,173,3,32,141,123,5,173,4,3
2,141,124,5,173,5,32,141,125
5,173,6,32,141,126,5,173,7,32,141,12
7,5,173,8,32,141,128,5,173,9
32,141,129,5,173,10,32,141,130,5,17
3,11,32,141,131,5,234,208,252
```

```
5   REM poke programma voor
    machinetaal
10  PRINT "CLR" : REM SCHERM
    SCHOON
20  S = 4096 : REM START ADRES
    MACHINETAAL
30  READ A
40  IF A = -1 THEN END
50  POKE S,A
60  S = S + 1
70  GOTO 30
100 DATA
    169,23,141,24,208,173,0,32,141
    ,120,5,173,1,32,141,121
110 DATA
    5,173,2,32,141122,5,173,3,32,1
    41,123,5,173, 4,32,141,124
120 DATA
    5,173,5,32,141,125,5,173,6,32,1
    41, 126,5,173,7,32,141,127
130 DATA
    5,173,8,32,141,128,5,173,9,32,1
    41,129,5,173,10,32,141
140 DATA
    130,5,173,11,32,141,131,5,234,
    208,252,-1
```

Je kunt zien dat we eerst een naam ergens in het geheugen hebben gezet met het Basic programma, wat normaal gesproken gebeurd als je het programma laadt. Daarna gaan we met een machinetaal programmaatje elke ASCII-waarde ophalen uit die geheugenlokatie en we plaatsen hem vervolgens in het videogeheugen. Het is natuurlijk vanzelfsprekend dat je de computer na het runnen van het eerste programma, (voor het poken van de naam) niet mag uitzetten, want anders verwijder je zojuist gePOKEte naam uit het geheugen.

Snel ?

In het hiervoor gegeven machinetaal programma staan weliswaar wel twee nieuwe instructies die er op dit moment echter niet veel ter zake doen. Deze instructies worden hier even gebruikt om een lus te maken, omdat uit het terug komen naar de shell of kernal het scherm wordt schoongemaakt, zodat het effect van ons programma verloren gaat. Om dit te voorkomen zetten we het machinetaal programmaatje in een lus zodat hij in machinetaal blijft lopen. Je kunt de computer onderbreken door te drukken op RUN/STOP RESTORE. Als je alles hebt ingetoets, blijikt het laatste Basicprogramma, hoewel het uit veel meer DATA-bytes bestaat, exact hetzelfde te doen als het allereerste programmaatje in deze aflevering. Een ook de uitvoer van dit programma gaat eerder trager dan de Basiclisting. Stel je echter gerust, dit is niet meer dan een omslachtig voorbeeld om de eerste werking van een dergelijke manier van programmeren uit te leggen en zeker geen standaardmanier om in machinetaal te programmeren. Daar hebben de heren ontwerpers gelukkig wel betere methoden voor bedacht.

RUN

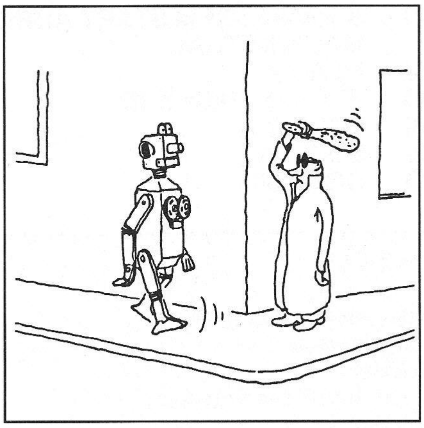
Voor het starten van het progamma kun je volstaan door het intypen van

SYS 4096. In het begin van het programma beginnen we door het ASCII nivo laag te maken, want doe je dat niet, dan krijg je alleen grafische karakters van de hogere ASCII. Voor de meer geavanceerde manieren hebben we eigenlijk een monitor of assembler nodig, omdat het anders een te lange invoer procedure wordt, daarom doen we de volgende keer dit experiment nog een keer, alleen dan met de monitor of assembler, maar omdat je waarschijnlijk nog niet eens weet wat de kreet monitor of assembler inhoudt, gaan we eerst wat dieper in op, wat je allemaal met een monitor en een assembler kunt doen, en wat voor voordelen dit heeft.

Monitors

In het begin van deze cursus zijn we begonnen met vertellen dat de processor werkt met nullen en enen dus bijvoorbeeld **1010 1101 0011 0101 0001 0100.** Dit zegt ons niets, maar voor de microproccessor is net zo duidelijk als voor ons is 1 + 1 = 2.
Maar aangezien wij de computer moeten gaan leren hoe we bepaalde problemen opgelost willen zien, moeten wij dus uiteindelijk een programma maken in de vorm zoals die hierboven staat. Dadelijk zul je zien hoe belangrijk nu een monitor of assembler is, omdat dat je daar veel sneller mee kunt programmeren, veel tijd mee bespaart, en omdat het schrijven in machinetaal toch al langer duurt dan een hogere taal als bijvoorbeeld Basic, kunnen we het ons gewoon niet permiteren om in nullen en enen te gaan schrijven. Zouden we dit wel doen, dan zouden de programma's zo duur worden, dat niemand ze meer kon kopen, wat dus zou betekenen dat we nog met de kroontjespen en het telraam op het buro de administratie zouden moeten doen, wat inmiddels allang verleden tijd is.
Het probleem hier is dat de **notatie** van het programma te lang en te ingewikkeld isen daardoor tijdrovendl; ook de kans op het maken van fouten behoeft geen nadere uitleg, terwijl het dubuggen van alleen maar series nullen en enen een ondoenlijk karwei is.

Voor dit soort problemen is het hexdecimaal getalstelsel in het leven geroepen om de binaire notatie te verkorten. Wat het verschil is tussen binair en hexadecimaal is voor ons geen enkel probleem meer, omdat wij dit tot treurens toe in vorige afleveringen geoefend hebben (als het goed is). Gaan we dan de binaire notatie van het vorige voorbeeld vertalen in een hexadecimale aanduiding dan krijgen we de notatie: **AD 35 14**. Hier kun je zien dat dit al vier keer zo kort is, en derhalve ook veel overzichtelijker. Dus als we op deze manier de programma's mogen invoeren ten opzichte van binaire invoer, zijn we al een stuk beter af.
Deze sprong kunnen we al verwezenlijken door het toepassen van een monitor In de vorige uitgave heb je kunnen zien hoe we een klein programmaatje maakten, en hoe we dit tenslotte in een klein voorbeeldje heben gezet. In dit voorbeeld hadden we de mnemonic code en de hexcode erbij gezet. De hexcode moesten we vervolgens omrekenen naar decimaal, omdat we anders deze hexadecimale waarden niet in het geheugen konden poken. Als je over een monitor beschikt dan kun je gelijk de hexadecimale waarden invoeren in een plekje in het geheugen. Een monitor in de zin zoals wij hem hier bedoelen is niet dat je er een televisie of een ander soort kijktoestel moet aanschaffen om in machinetaal te kunnen programmeren.



Monitor betekent "*het laten zien*", en dat is precies wat we er mee gaan doen. De **monitor** is een programma, waardoor we in staat gesteld worden om de inhoud van een stuk geheugen zichtbaar te maken, in hexadecimale vorm, dus niet in nullen en enen waar we toch niets van begrijpen. We kunnen op deze wijze ook rechtstreeks hexadecimaal invoeren in het computergeheugen. Dit heeft al veel meer voordelen ten opzichte van het poken in Basic. In de monitor kun je een ingevoerd programma gelijk testen door het even te runnen. Als het programma doorlopen is komt de computer gewoon terug in de monitor, en kun je eventueel een wijziging aanbrengen als je iets fout gedaan hebt. Maak je een fundamentele fout, zoals het verkeerd uitvoeren van bijvoorbeeld een sprong-opdracht, dan is het logisch dat de computer zich ophangt (vastloopt) en niet meer terug komt in de monitor.

Als je het programma nog een keer wilt overzien, dan kun je gewoon het stuk geheugen listen waar je het programma hebt geplaatst. Stel dat je een programma hebt ingevoerd wat begint op $ 1000, dan voer je in '**.m 1000 1050**' Na dit bevestigd te hebben met de enter toets drukt de computer in dit geval de inhoud af van geheugen lokatie $ 1000 tot $ 1050. Het '**.m**' is dus een typisch monitor commando, net als bij Basic 'LIST'. Op deze wijze kun je dan gelijk controleren of je wat verkeerd hebt ingevoerd.

Assemblers

Vervolgens kunnen we nog een stap nemen voor het nog gemakkelijker invoeren van een programma. Deze stap is het gelijk invoeren van de **mnemonic code** in plaats van het intypen van opcodes en operands. Deze aanvulling op de monitor noemen we de **assembler**. Assembler betekent 'in elkaar zetten'.
Het voordeel van de assembler ten opzichte van de monitor is dat je niet meer zoveel in het boekje met de instructieset hoeft te kijken, omdat je de instructie gelijk aan de assembler in mnemonic opgeeft. In het onderstaande voorbeeld zetten we een testje in monitortaal geschreven en een voorbeeld in assembler geschreven.

Voorbeeld Monitor:

: AD002069 058D0030
: 00

Voorbeeld Assembler:

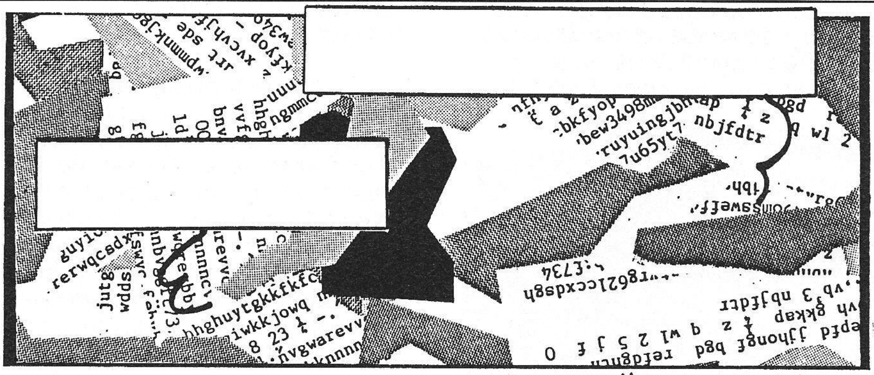
LDA $2000
ADC #$05
STA $3000
BRK

Je kunt zien dat er een heel verschil is tussen de monitor en de assembler. Stel dat je in de monitor het getal $5 in de accumulator wilt laden, dan moet je in de instructieset eerst de instructie LDA opsporen, vervolgens achter de bedoelde instructie de juiste instructiemethode opzoeken, in dit geval immediate. Achter de instructie LDA komen de volgende instructiemethoden voor, met de daarbij behorende code:

immediate	$A9,
absolute	$AD,
zero page	A5,
indirect X	$A1,
indirect Y	$B1,

zero page X	$B5,
absolute X	$BD,
absolute Y	$B9,
indirect	$B2.

Zo zie je dat achter een simpele instructie LDA 9 verschillende code's voorkomen, waar je dan een keuze uit moet maken. Van deze 9 code's hebben we er inmiddels 3 behandeld namenlijk de immediate, absolute en de zero page methodes. Dus in dit geval kiezen we voor de code $A9 met het getal $05 erachter, dus 'A9 05'. Willen we dit getal ergens uit het geheugen inlezen, bijvoorbeeld absolute vanaf geheugen-lokatie $7345 dan krijg je de code $AD met daarachter het geheugen adres $7345, dus 'AD 45 73'.



„ DATA-DIALOOG "

In assembler gaat dit alles veel eenvoudiger namelijk: hier wordt LDA #$05 ingetypt voor immediate-mode, en voor de absolute LDA $7345.
Het verschil tussen immediate en absolute geven we hier aan door middel van het tekentje #, denk maar eens terug aan het eerder gegeven voorbeeld waar dit ook ter sprake kwam. Dit voorbeeld even ter illustratie om het verschil aan te geven tussen monitor en assembler, want wij gaan de volgende keer door met het programmeren in de assembler. Wat ons hieruit wel duidelijk moet zijn geworden is dat je hoeft niet meer uit 9 verschillende code's te kiezen, met het daaraan verbonden risico een foute keuze te maken.

Debuggen

Door het programmeren in assembler wordt het allemaal wat overzichtelijker en je kunt er echt een programma in herkennen, wat natuurlijk voordelen heeft als je moet gaan debuggen voor het opzoeken van een gemaakte fout. Debuggen betekent ontleden, of eigenlijk ontluizen. Maar omdat er gelukkig niet veel van dit ongedierte in de computer voorkomt houden wij het voorlopig maar even op ontleden. Dit ontleden slaat dan op het programma, en niet op de computer. Stel dat je bij het invoeren van het programma een typefout maakt, dan is de kans dat je in de monitor de fout ontdekt heel klein.
Als je hem kan debuggen in de assembler, dan zie je echt het programma voor je en kun je gemakkelijker de fout eruit halen. We nemen het vorige voorbeeld om dit te illustreren, waar we een klein foutje in hebben gezet.

Voorbeeld Monitor:

: **AD002069 058D0330**
: 00

Voorbeeld Assembler:

LDA $2000
ADC #$05
STA $3003
BRK

Vergelijk deze listings met de vorige en kijk of je de consistente fout eruit kunt halen. Het zal daarbij onmiddelijk opvallen, dat dit veel gemakkelijker gaat in de assembly-listing dan in de listing van het monitorprogramma, nog daargelaten hoe moeilijk het is in een binaire uitdraai van dit toch niet al te grote programma.

Voordelen

In principe heeft het werken voor ons met de monitor niet veel zin, omdat we al behoorlijk hebben kunnen oefenen op hexadecimaal niveau, daarom kiezen we ook voor de assembler, omdat we dan wat sneller de andere instructies met de daaraan verbonden methodes kunnen doornemen. Het is wel belangrijk om de hiervoor behandelde stof goed te onthouden, want je blijft het altijd nodig houden. Deze depressieve toelichting houdt niet in dat dit het einde zou betekenen op het toepassen van vereenvoudigingen op assembler nivo. Dit is gelukkig niet waar want er is een manier om nog effectiever te programmeren, en dat heet '**macro assembler**'. Dit is iets waar wij voorlopig nog niet mee gaan werken, omdat wij het eerst goed met de zogenaamde mini assembler afkunnen, ook omdat we anders te veel zaken in één keer vastpakken, want de macro assembler heeft een commando set op zichzelf.
Toch geven we even aan wat de mogelijkheden zijn. Bij de **(mini)assembler** wordt de ingevoerde opcode + operand direct in het geheugen geplaatst, dus als je klaar bent met invoeren hoef je alleen maar het commando **.g $ 'startadres'** te geven en het programma werkt, tenminste als je alles goed ingevoerd hebt.
Bij **macro assembling** maak je eerst een programma in een tekst editor (Een tekst verwerker), en daarna ga je die **compilen** (overzetten in machinetaal door middel van een speciaal programma)

In deze tekst editor hoef je niet meer te werken met hexadecimale getallen, maar met **labels**. Je hoeft alleen maar in het begin een getal toe te kennen aan ieder label wat gebruikt wordt. Bijvoorbeeld het startadres = $ 1000 ken je toe aan de naam START. Je hoeft dan niet meer te springen naar een adres, maar naar een naam, die door de compiler wel wordt vertaald in een getal. Ook het tussenvoegen van een routine gaat hier heel eenvoudig, wat bij de mini assembler heel lastig is. Maar hier komen we wel op terug als de gelegenheid zich voordoet in de vorm van een voorbeeld. Hopenlijk is het je een beetje duidelijk geworden wat het verschil is tussen een "monitor", "assembler", en "macro assembler".

De volgende keer plaatsen we, als alles goed gaat tenminste, een **listing van een monitor/assembler** in het blad, die je zelf over kan tikken, en opslaan op tape of disk. We gaan dan een beetje spelen met de assembler om er wat vertrouwd mee te raken. Eerst maar weer veel succes en tot de volgende keer.

Tj.v.d.L.

Op het gebied van de huiscomputers is Commodore al lang één van de grootste bedrijven in de wereld. Ook in de IBM-compatibele zakencomputers gaat het niet slecht met het bedrijf. Aan de onderkant van het scala PC's is nu eindelijk de PC-1 op de markt gekomen.

De Commodore PC-1

We hebben er lang op gewacht, op die kleine PC van CBM. Tegelijk met het goed nieuws van een redelijke winst (28,6 miljoen dollar), op een omzet van 806,7 miljoen dollar, kwam ook de 'low-end' machine uit.

De nieuwe instapper van Commodore voor de MS-DOS markt is duidelijk geen uitgeklede PC-10 geworden. Men heeft een aantrekkelijke kast gemaakt, er een behoorlijk compleet electronisch hart ingezet en de hele zaak zal ook vast wel een aantrekkelijke prijs krijgen. Die is nog niet precies bekend, maar in de standaardconfiguratie, met één diskdrive, een monitor en wat software zal het wel rond de tweeduizend gulden gaan worden.

De opvallendste karakteristiek van de PC-1 zijn de buitenafmetingen, de systeemkast meet maar 33 x 32 x 8,5 centimeter.

In de PC-1 zit als processor een 8088 microprocessor van Intel, die op de vrij trage oorspronkelijke PC snelheid van 4,77 MHz draait. Er zit 512 KB RAM geheugen in, op de moederkaart uit te breiden tot 640 KB, dat is dus het maximum dat met de gewone MS-DOS kan worden benut.

Vooral op grafisch gebied heeft Commodore meteen maar een multi-mode videokaart gemonteerd, men zal met de PC-1 niet snel vastlopen, omdat die kaart zowel monochroom (MDA), kleur (CGA) als Hercules compatibel is. Dat wil dus zeggen dat men het beste op videogebied combineert in een kaart en de gebruiker dus zowel spelletjes, tekst als grafische programma's in bit-map mode als Ventura zal kunnen draaien.
Verder heeft men ook op interface gebied alles standaard ingebouwd, namelijk zowel een seriële als een parallel interface voor de printer en/of modem plus een muis-interface.



Wie nog meer wil doen en dan denken we meteen aan een harde schijf, zal buiten het (kleine) kastje verder moeten uitbreiden. Daartoe is er een speciale uitbreidingsconnector (geen normale PC sleuf) en komt er van Commodore een uitbreidingskastje met wel normale sleuven.

Het toetsenbord telt 84 toetsen, met de functietoetsen opzij, hetgeen in de praktijk toch prettiger werkt dan functietoetsen op een rij bovenaan. De enkele diskdrive kan 360 KB aan data bevatten en het geheel werkt onder DOS 3.2. Er wordt naast de MS-DOS ook een GW-Basic meegeleverd. Wie een andere diskdrive wil hebben, kan ook een 3,5 inch eenheid met 720 KB inbouwen in plaats van de normale diskdrive.

Voor de Commodore 64 al jarenlang een goede bekende, nu ook voor de Amiga beschikbaar: VizaWrite.

Vizawrite voor de Amiga

Nieuw op de Nederlandse Amiga markt is het teksverwerkingspakket VizaWrite. Deze naam is zeker geen onbekende bij de grote schare Commodore 64 en 128 gebruikers. Het pakket staat alom bekend om zijn eenvoudige opzet. Zelfs als men nog nooit met een tekstverwerker heeft gewerkt, kan men hiermee na een korte periode probleemloos werken. Commodore heeft met de Amiga 500 en 1000 een behoorlijk stuk van de hobbymarkt veroverd. De 2000 begint nu ook langzaam door te dringen tot de bedrijven.VizaWrite heeft voor deze beide sectoren zijn aantrekkelijke kanten.

Tekstverwerken

Teksten moeten door vele mensen verwerkt worden. Bedrijven, verenigingen en particulieren allemaal hebben we er mee te maken. Het vroegere gezwoeg op een typemachine begint langzaam maar zeker tot het verleden te behoren. Het vervelende bij de typemachines was dat er altijd fouten werden gemaakt, of correcties moesten worden door gevoerd. Om tot een mooi resultaat te komen moest het geheel dan worden over getypt. 1et de komst van tekstverwerkers behoort dit alles tot het verleden. Fouten zijn zo verholpen, indelingen zijn te veranderen, zelfs staten maken is kinderwerk geworden. Een van de grootste voordelen van een tekstverwerker is waarschijnlijk dat er stukken tekst kunnen worden bewaard. Op elke gewenste plaats zijn deze weer door middel van een enkele handeling te gebruiken. Op veel bedrijven wordt bij de correspondentie alleen maar gebruik gemaakt van kant en klare tekstblokken. Snelheid en kwaliteit gaan hier dus samen.

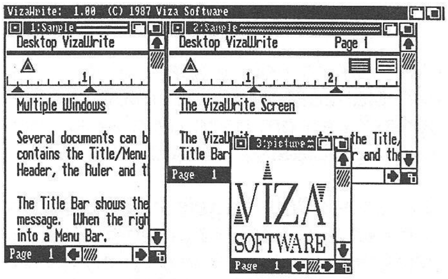
VizaWrite.

Het Vizawrite-pakket maakt volledig gebruik van de beroemde Amiga stijl. Alle mogelijk heden van de computer worden gebruikt.
VizaWrite maakt gebruik van de muis. Vele handelingen die normaal bijvoorbeeld door codes moesten worden, ingevoerd zijn nu door middel van een handeling met de muis uit te voeren. De meervoudige raamwerken maken het mogelijk aan meer documenten tegelijk te werken.
Wat is er nodig om met VizaWrite te werken.Aan hardware, dus aan apparatuur:

° Een Commodore Amiga met 256 Kb geheugen. Dit is de kleinste geheugencapaciteit waarmee een Amiga wordt geleverd, dus het pakket op alle Amiga's.
° Minimaal één diskdrive, ook dit is standaard dus geen probleem.
° Een monitor welke geschikt is voor 80 karakters en een Amiga aansluiting. Het mooiste is echter een kleurenmonitor, een monochrome monitor werkt echter ook probleemloos.
° Een printer die door VizaWrite ondersteund wordt en de juiste aansluitkabel om deze aan de Amiga aan te sluiten.

Standaard in het pakket zijn de volgende printers: Mps 1000, Mps 2000, Diablo 630, Epson FX-HP laserjeten de Juki 6000 met een PS optie. Door de Preference optie te kiezen is bijna elke printer via de printer set-up te gebruiken.

De software bestaat uit:



° Het VizaWrite pakket. Voor het opstarten van het programma is het veiliger eerst een kopie hiervan te maken en het orgineel op een veilige plaats op te bergen. (Denk er aan dat dit alleen mag voor eigen gebruik).
° De Amiga Kickstart disk. In sommige Amiga computers is deze ingebouwd in de ROM. De kickstart versie V1.1 zowel als V2.1 is te gebruiken.
° Een of meerder geformatteerde diskettes om de documenten op weg te schrijven.

Het gebruik

Het VizaWrite pakket benut volledig alle voordelen van de Amiga zoals het op ikonen gebaseerde operating systeem, met een menu boven in het scherm en meerdere raamwerken. Als u nog niet thuis bent in het werken met de Amiga is het verstandig eerst wat te oefenen met de verschillende Amiga mogelijkheden. Gebruik hiervoor de Kickstart en de Workbench disk. U kunt ook de workbench van VizaWrite gebruiken. Als u hiermee overweg kunt is het werken met VizaWrite geen enkel probleem meer. Het is een goede gewoonte alle overbodige windows te sluiten dit om geheugen ruimte te sparen. Aanzetten van de computer en het 'booten' zoals u gewoon bent is voldoende om het workbench-scherm te laden. Met de muis wordt nu het VizaWrite ikoon tweemaal snel achter elkaar aangeklikt en het programma wordt nu geladen. Het is ook mogelijk om via de ikoon documenten direkt te laden.

Eventuele instellingen, bijvoorbeeld de kleuren, zijn via de preferences eenvoudig te veranderen. Hierdoor is het geheel naar eigen wensen aan te passen. Na het laden van het programma VizaWrite verschijnt het edit scherm.
Dit is door het aanklikken van het zwarte vlakje rechsboven naar de achtergrond te verplaatsen. Zo komt u dan weer in het workbench-scherm terecht. U kunt nu bepaalde instellingen veranderen zonder dat het programma waar u aan bezig bent verlorten gaat. U keert weer terug naar het edit scherm door het aanklikken van het witte blokje rechtsboven. Een tweede

mogelijkheid is het scherm niet naar de achtergrond te verplaatsen, maar naar beneden te trekken. Dit gebeurt door de gestreepte balk aan de bovenkant aan te wijzen en de knop van de muis ingedrukt naar beneden te trekken. Documenten welke worden weggeschreven naar diskettes, krijgen een eigen ikoon, die lijkt op een beschreven vel papier. Bij het dubbel aanklikken van deze ikonen wordt eerst het programma VizaWrite geladen en meteen het bijbehorende document.
VizaWrite heeft geen aparte workbench nodig om op te starten dus kan het zonder problemen op de harddisk worden gezet. Omdat het pakket kant en klaar wordt gelevert is er na het opstarten van het programma direkt te beginnen met het intypen of bewerken van de teksten.

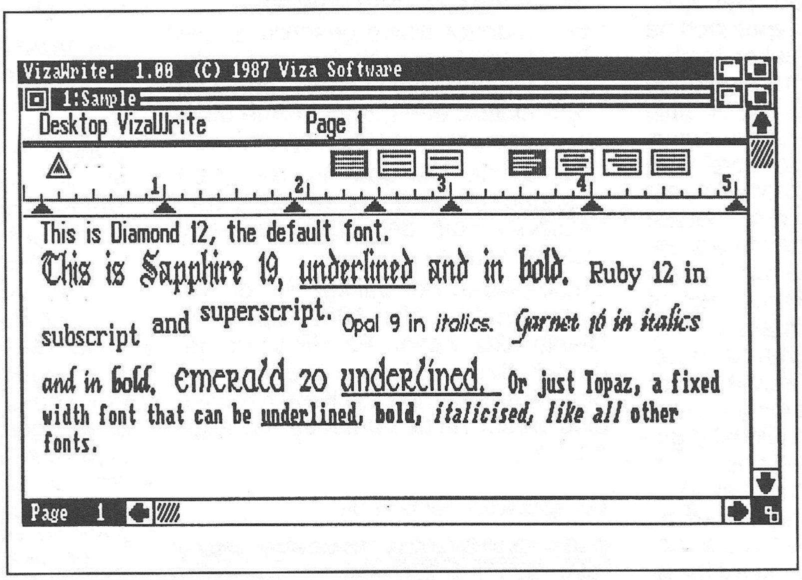
Muisbesturing

Voor de mensen die veel met een tekstverwerker omgaan,is het misschien wennen om alle funkties met de muis te besturen. Alle opdrachten die met de muis werken zijn ook via het toetsenbord te gebruiken. Dit heeft het voordeel dat een ervaren persoon bepaalde opdrachten die veel voorkomen sneller kan uitvoeren.
Door met de muis de menubalk, boven in het scherm aan te wijzen en de rechterknop ingedrukt te houden verschijnen alle besturingsfunkties. Hieruit kan men er een kiezen door de muis naar beneden te trekken.
De zwarte balk vertelt welke funktie er gekozen is.

Menu

Het menu is verdeeld in zeven hoodfunkties. Het file-menu geeft alle mogelijkheden om een document op te vragen, te sluiten, te verwijderen, te printen, de printer te wijzigen, maar ook informatie over het dokument is hiermee op te vragen; wie het geschreven heeft, op welke dag en tevens worden alle statistische gegevens op het scherm getoond. Het edit menu geeft de mogelijkheden blokken te kopieeren, te verwijderen, lijnen te plaatsen en pagina koppen en voetnoten te plaatsen. Zoek en vervang funkties bevinden zich in het zoekmenu. Hiermee kan men zowel zoeken naar een woord als naar een of meerdere letters. Zoeken en vervangen kan alleen vooruit. In het Fontmenu zijn er standaard 7 verschillende lettertypes opgenomen die allemaal in de verschillende groottes kunnen worden weerg egeven. Waarbij alleen Topaz een niet-proportionele letter geeft. In het style menu kan men kiezen of de lettertypes onderlijnd, dik, in subscript, in supersript of schuin moet worden afgedrukt.

Met behulp van het Imagemenu zijn er tekeningen uit de diverse tekenpakketten tussen de tekst in te laden. Bijvoorbeeld Graficraft of het beroemde DeLuxe Paint pakket. Deze tekeningen zijn door middel van een raam werk probleemloos te vergroten, te verkleinen en te verplaatsen. Hierbij past de tekst zich volledig aan. De tekening is er dus gewoon tussen te plaatsen. In het glossary menu kunnen standaard stukken tekst worden opgeslagen om deze op diverse, willekeurige plaatsen te gebruiken.



Dit kunnen tekstregels zijn maar ook komplete adressen zijn mogelijk. Een extra mogelijkheid van dit programma is de mogelijkheid van Mailmerge. Op willekeurige plaatsen in het document zijn er invoegtekens op te nemen ( ) waarvoor dan automatisch bepaalde teksten worden afgedrukt. Deze teksten worden in een datafile opgeslagen. Deze zijn zowel met een Database te maken maar ook met het VizaWrite pakket kunnen deze worden aangemaakt. Door middel van PrintMerge kunnen deze gecombineerde teksten worden uitgeprint.

Voordelen

- VizaWrite is zeer algemeen, teksten uit bijvoorbeeld Textcraft, tekeningen uit de LuxePaint, Graficraft het is allemaal geen probleem. Editten en saven van alle Ascii teksten is mogelijk.
- Het werkt zowel met de NTSC als de PAL computers
- Teksten worden automatisch in pagina's ingedeeld.
- Wat er op het scherm staat,komt op de printer.
- Kop en voetregels worden op alle pagina's getoond en kunnen uit meerdere regels bestaan.

Indelen van de Lay)out, kantlijnen, tabs, uitlijnen en de lijnspati37ring gebeurt op een eenvoudige wijze met de muis; de verandering is op het beeldscherm te volgen.
Tussentijds veranderen van de lettertypes is altijd mogelijk en op het scherm zijn deze wijzigingen te zien.

Verwijderen en kopieeren gaat zeer snel omdat er met teksblokken kan worden gewerkt welke door de muis worden aangewezen. Ze zijn herkenbaar door een felle oranje kleur.
Scrollen is in alle richtingen mogelijk. Alle internationale karakters kunnen worden gebruikt.
Er kan worden gewerkt met verschillende defaults voor printer, diskdrives en documenten.

Nadelen

Het enige nadeel dat ik heb ondervonden bij het werken met dit pakket is dat het niet mogelijk is diskettes vanuit het programma te kunnen formatteren.
Het VizaWrite pakket is ontwikkeld door Kelvin Lacy, ook de auteur van de C64 en 128 Versie. Het copyright berust bij Viza Software Limited. Het pakket wordt in Nederland op de markt gebracht door de firma: Altycos uit Zoetermeer. Voor inlichtingen kunt u bellen: 079-510757.
De prijs voor deze uitstekende tekstverwerker is in vergelijking met andere pakketten pakketten zeer laag gehouden *f* 259,50 (incl. btw).

Vanaf nu zal de Miniatuurtjes-rubriek er anders uitzien, een andere naam hebben: Basic Micro's, en ook een andere samensteller. Nadat Nico Baaijens bijna 30 keer deze rubriek verzorgde en in die tijd zelf als hoofdredacteur van PC World zijn stempel op de PC ontwikkeling drukte, volgt Peter Broekhuizen hem op als degene die de selecties maakt en kijkt of de programma's ook echt draaien.

Basic Micro's

In deze rubriek zullen we proberen zoveel mogelijk variaties aan te brengen wat betreft de programmaatjes. Niet alle programma's zullen altijd je interesse hebben, maar we proberen ook rekening te houden met mensen die nog nauwelijks programmeerervaring hebben, en voor wie een eenvoudig programmaatje met een leuk effect zeer verhelderend kan zijn.

We zullen verder zoveel mogelijk opdrachten op één regel proppen zodat er meer Micro's op de twee bladzijden passen. Erg lange programma's zullen worden doorgegeven aan de mensen die de listingrubriek verzorgen. Af en toe zullen we zelf ook een bijdrage leveren in de vorm van een kort programmaatje, verder kijken we bij elk ingezonden programma of het nog valt te verbeteren of te versnellen.
Tenslotte zullen voortaan alle Micro's in een niet-proportionele letter gedrukt worden, dat wil zeggen dat een 'i' even breed is als een 'w'. Dit kan het overtypen wat gemakkelijker maken.

Typemachine

De eerste Micro komt van Remco Bosshard uit Vlaardingen en maakt van je printer en je computer samen een typemachine. Alles wat je intikt, verschijnt op het scherm, en wanneer je op RETURN drukt zal de hele regel naar de printer gaan:

```
10   OPEN1,4
20   GETA$:IFA$=""THEN20
30   PRINTA$;:IFA$=CHR$(20)THENB$=LEFT$(B$,
     LEN(B$)-1AND255):GOTO20
40   IFA$=CHR$(13)THENPRINT#1,B$:B$="":GOTO20
50   B$=B$+A$:GOTO20
```

Pas op met toetsen als SHIFT-CLR/HOME, die wel effect hebben op het scherm, maar niet op datgene wat geprint wordt. Zelf uitbreiden dus, als je daar problemen mee krijgt.

Aandachttrekker

En nu een "teksteffect dat de aandacht trekt", zoals Bas Vinken uit Enschede het noemt. We hebben het zelf nog een beetje uitgebreid:

```
10   DIMA(14):PRINTCHR$(147):Q=1476:FORI=1TO14:
     READA(I):NEXT
20   FORS=0TO9:FORI=1TO14:POKEQ+I,A(I):NEXT:
     IFS<9THENFORI=1TO14:POKEQ+15-I,32:NEXT
30   FORTI=1TO14:N=15-I:POKEQ+N,A(N):NEXT:
     IFS<9THENFORI=1TO14:POKEQ+I,32:NEXT
40   NEXT:FORS=0TO9:FORI=1TO14:POKEQ+I,32:
     FORT=0TO4:NEXT:POKEQ+I,A(I):NEXT
50   FORI=1TO14:N=15-I:POKEQ+N,32:FORT=0TO4:
     NEXT:POKEQ+N,A(N):NEXTI,S:GOTO20
60   DATA 3,15,13,13,15,4,15,18,5,45,9,14,6,15
```

Simon's Basic

Simon's Basic blijkt nog steeds populair te zijn, gezien het feit dat hiervoor elke keer weer een flink aantal programmaatjes ingezonden wordt. Bernard Hudepohl uit Lelystad stuurde er maar liefst zeven tegelijk in, maar onderstaand programma was verreweg het beste. We hebben hem iets sneller gemaakt, door onder andere de berekening van COS(X) en SIN(X) voor de FORY=... te zetten; dit zijn details waar je soms een grote snelheidswinst mee kunt bereiken.

```
10   COLOUR0,0:A=10:B=2.4:Z=π/180:HIRES1,0:
     MULTI1,6,14
20   FORXX=0TO170STEP10:X=XX*Z:C=COS(X):
     D=SIN(X):E=D/B:F=2*A*C:J=J+1:IFJ=4THENJ=1
30   C=C/B/2:D=A*D:FORY=0TO360STEP10:
     G=SIN(Y*Z): A1=80+C*(Y-180)+D*G
40   A2=100-E*(Y-180)+F*G:IFYTHEN-
     LINEA3,A4,A1,A2,J
50   A3=A1:A4=A2:NEXTY,XX:PAUSE1
60   MULTI0,0,7:FORC=1TO50:NEXT:MULTI7,0,0:
     FORC=1TO50:NEXT
70   MULTI0,7,0:FORC=1TO50:NEXT:GOTO60
```

Willekeurige cirkels

Eindelijk eens iets voor de C-128! Het is een vrij eenvoudig proramma dat tot in het oneindige willekeurige cirkels tekent. Als je geen kleurenmonitor hebt, kun je de COLOR-opdrachten in regel 10 wel weglaten. Koen de Lille uit Damme-Sijsele (België) is de schrijver:

```
10   GRAPHIC1,1:COLOR0,1:COLOR4,1:
     COLOR1,INT(RND(1)*16)+1
```

```
20   C=INT(RND(1)*80)+1:D=INT(RND(1)*80)-1
30   CIRCLE1,150,100,C,D:CIRCLE1,150,100,D,C:
     GOTO 20
```

Lang zal ze leven

Voor verjaardagen zeer bruikbaar is de Micro van Roel Nikkessen uit Bergen. Je moet natuurlijk nog steeds zelf zingen, maar als je C-128 de wijs al voor je speelt, wordt "Lang zal ze leven" een makkie.

```
1    PLAY"QFIFSFQFQCQA.IASAQAQFO5QC.
     ICSCIDICO4$IBIAQG.IGSGQGO5ICO4$IB
     HA$HBO5HCQDO4$QBHAHGHF"
```

De taal van de machine

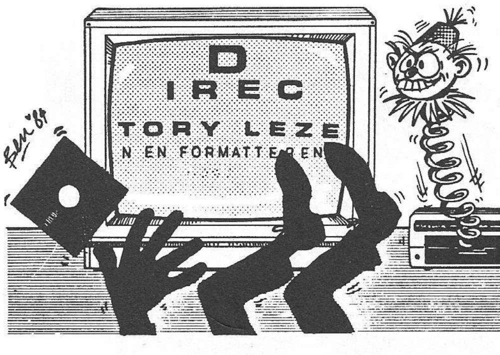
Machinetaal is natuurlijk ook een onderdeel van deze rubriek.
Ondanks het grote aantal programma's in deze rubriek dat al vaak voor flitsende schermen heeft gezorgd, volgen er hier weer enkele voor de C- 64, want leerzaam blijft het. Met dank aan PAS (Pascal's Advanced Software):

```
10   FORI=8192TO8231:READA:X=X+A:POKEI,A:NEXT:
     IFX<>5329THENPRINT"FOUT IN DATA":END
20   DATA169,0,141,32,208,141,33,208,162,0,232,
     236,0,48,208,250,169,1,141,32,208
30   DATA141,33,208,166,0,232,236,0,48,208,
     250,169,64,205,197,0,240,217,96
40   INPUT"[SHIFT-CLR/HOME]SNELHEID";A:
     POKE12288,A:WAIT203,64:SYS8192:GOTO40
```

Kleurenbalken

Kleurenbalken kun je volgens Obbe Vermeij uit Capelle aan de IJssel op de volgende manier programmeren:

```
10   FORX=0TO14:READY:POKE49152+X,Y:
     NEXT:INPUT"GETAL (0-255)";G:POKE49160,G
20   POKE53280,1:POKE53281,1:SYS49152
30   DATA120,238,32,208,238,33,208,162,
     229,202,208,253,76,1,192
```



Terughalen

Nu geen echt programma maar een methode om een programma terug te krijgen dat per ongeluk een NEW of RESET voor z'n kiezen kreeg, en dit alles van Martin v.d. Plas uit Katwijk aan Zee, voor de C-16:

```
     POKE4097,28:POKE4098,16:POKE45,85:POKE
     47,85:POKE49,85
```

Grafisch grapje C-16

En dan ook nog maar wat grafische grapjes voor de C-16, van Michaël Cramers uit Begijnendijk (België):

```
10   GRAPHIC1,1:FORX=4TO160STEP2:
     CIRCLE1,0,0,1.5*X,X,90,90+X/2,270
20   CIRCLE1,319,0,1.5*X,X,270-X/2,270:NEXT

10   GRAPHIC1,1:FORX=0TO360STEP6:
     CIRCLE1,65+X/2,80,60,33,,,X:NEXT
```

Verschrikkelijke border-verdwijning

Voor hen die de machinetaal vrezen volgt hier een echte verschrikking. Boris van Woerkom (14 jaar) uit Oosterhout is de maker van deze Micro voor de C-64. Gelukkig heeft hij er een controle-teller ingebouwd. Het programma zorgt dat de border verdwijnt, zodat sprites tot aan de rand van je beeldscherm zichtbaar blijven.

```
10   FORT=49152TO49241:READA:B=B+A:POKET,A:
     NEXT:IFB=11198THENSYS49152:END
20   PRINT"FOUT IN DATA!!":
     DATA169,44,141,20,3,169,192,141,21,3,169,0
     ,141,89,192
30   DATA141,14,220,169,127,45,17,208,141,17,
     208,169,60,141,18,208,169,129,141
40   DATA26,208,173,25,208,141,25,208,88,96,
     169,1,141,25,208,172,89,192,185,85
50   DATA192,141,18,208,185,87,192,234,234,141,
     17,208,200,192,2,208,8,160,0,140
60   DATA
     89,192,76,49,234,140,89,192,76,188,254,
     249,255,27,19,1,61,0,36,61,0,0,0
```

Voor romantici

Voor de romantici onder ons, die vinden dat Nederland en België 's nachts veel te vaak dicht bewolkt zijn, volgt hier een sterrenhemel, ontworpen door Francis Vandewalle uit Sijsele (België).

```
1    PRINTCHR$(147):POKE53281,0:FORI=1TO50:
     POKE1024+RND(0)*1000,46:NEXT:WAIT0,0
```

C-16 en C-128

Veel mensen klagen dat er zo weinig aandacht is voor de C-16 en de C- 128. In deze rubriek proberen we hier wat aan te doen, maar we zijn hierbij wel afhankelijk van de ingezonden programma's. Bij deze wil ik iedereen daarom nog eens uitdrukkelijk uitnodigen om eigen gemaakte programma's in te zenden.
Als laatste een tip voor bezitters van de C-16 en C-128 computers. Sommige opdrachten werken exact hetzelfde op beide apparaten, dus als er een Micro voor de C-16 in staat, probeer hem dan ook eens op je C-128, of omgekeerd. Doe dit echter niet als er POKE-opdrachten instaan!

Het tachtig-koloms scherm van de Commodore 128 kan een resolutie van 640 x 200 punten bereiken. De bit-map instructies voor het 40-koloms scherm werken echter niet. Daarvoor heeft Peter Heinckiens in de eerste aflevering van dit artikel een programma ontwikkeld, Minigraph. Dat programma wordt in deze aflevering verder uitgewerkt.

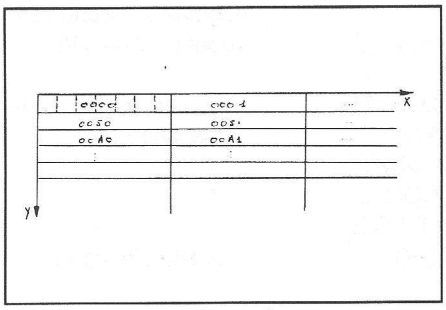
Hoge resolutie met het 80-kolomscherm

(deel 2)

We hebben in de eerste aflevering (3e jaargang nr. 9, dec. 1986) de methode uiteengezet om te werken met het hoge-resolutiescherm in 80-kolomsmode. In dit nummer gaan we het toen ontworpen programma, MINIGRAPH, verder uitbreiden.

Hardcopy

Indien u over een printer beschikt, zult u vast wel eens een afdruk van uw grafische kunstwerkjes willen maken. Daarom willen we u de volgende hardcopy-routine (voor de MPS 803) zeker niet onthouden. Maar een beetje uitleg is daarvoor wel nodig.
We hebben in aflevering 1 gezien hoe het hires-beeldscherm in de computer wordt opgeslagen, nl. volgens het volgende schema:



schema 1.

Daar de MPS 803 vertikaal 7 naalden heeft, is hij in staat 7 rijen ineens af te drukken. Horizontaal heeft deze printer slechts 1 naald, dus kan er per 'aanslag' slechts 1 kolom worden afgedrukt. Dit alles wordt gedaan door de volgende byte te vormen:

```
1 x x x x x x x
  !     ! ! !
  rij 7 ! ! rij 1
        ! !
        ! rij 2
        !
        rij 3
```

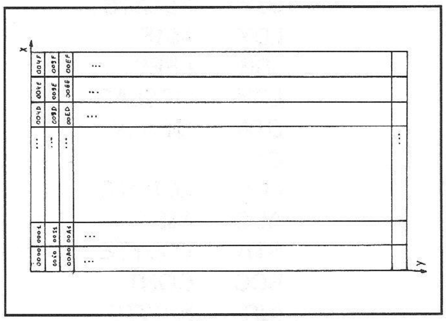
waarbij bit 7 steeds geset is, en de overige bits 0 of 1 zijn, naargelang de overeenkomstige bit in de overeenkomstige rij respectievelijk 0 of 1 is.
Laten we dit illustreren aan de hand van een voorbeeld:
Stel dat we de volgende figuur willen printen:

```
********
*
 *
  *
 *
*
********
```

Dan dienen we de volgende data naar de printer te sturen:

kolom 1 : 11100011 (bit 7 is altijd geset)
kolom 2 : 11010101
kolom 3 : 11001001
enz.

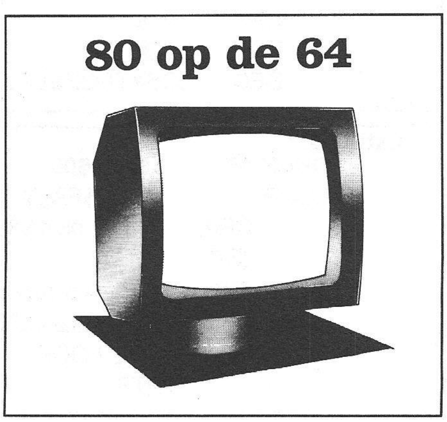
Nu doet zich wel een probleem voor: de MPS-803 heeft slechts een horizontale resolutie van 480 punten, terwijl het beeldscherm horizontaal 640 punten heeft. Bijgevolg zullen we genoodzaakt zijn om het scherm in de lengterichting af te drukken.
Ook de VDC zorgt voor de nodige moeilijkheden: door zijn langzame en omslachtige adressering, zou het veel te lang duren indien we rechtstreeks met het schermgeheugen moesten werken. Een alternatief zou kunnen zijn het volledige schermgeheugen naar de RAM te kopieren, maar dit zou 16K innemen, wat toch betrekkelijk



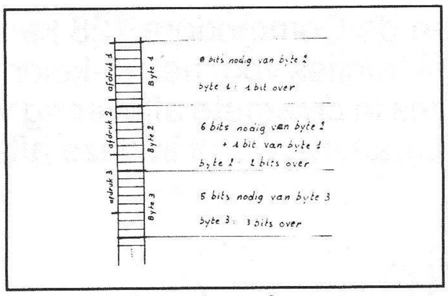
schema 2.

veel is. De oplossing: kopieer de VDC-RAM in kleine delen tegelijk, en druk deze elk afzonderlijk af.

Zoals reeds gezegd, moeten we het scherm 'op zijn kant' afdrukken, dus volgens schema 2.
We zullen 1 schermkolom tegelijk kopieren en afdrukken, en we beginnen met de laatste kolom. Dit dienen we 92 maal te herhalen.
Alsof er nog niet genoeg waren, duikt er weer een probleem op: een Byte heeft acht bits, en houdt dus infor-



matie vast van acht pixels. De MPS-803 kan echter slechts zeven punten tegelijk afdrukken, en bijgevolg blijven we met een bit zitten. Deze bit zal nu met de volgende Byte samengenomen moeten worden, met als resultaat dat we nu twee bits overhebben. Eén en ander wordt verduidelijkt in schema 3.



schema 3

Hoe gaan we nu te werk? Noemen we N het aantal bits die nodig zijn van de volgende Byte, dan zien we dat N achtereenvolgens de waarden 0, 6, 5, 4, 3, 2, 1, 0, 0, 6, 5, 4, ... aanneemt. 0 wordt dus tweemaal bereikt: één keer voor bits 0 tot en met 6 en eenmaal voor bits 1 tot en met 7. In het eerste geval zullen we onmiddelijk de accumulator kunnen laden met de Byte-waarde, en deze naar de printer sturen (na er eerst $80 bij opgeteld te hebben), in het tweede geval echter,

listing 1

```
KOPIEER  LDA  LOBYTEOPSLAG; we plaatsen de
         STA  LOBYTE      ;pointer LO/HIBYTE
         LDA  HIBYTEOPSLAG ; op de af te drukken
         STA  HIBYTE      ; lijn
         LDY  #$00        ; kolomteller op nul
LOOP     STY  OPSLAG      ; en bewaren
         LDY  #$12        ; zet pointers op de in
         LDA  IBYTE       ; te lezen byte
         JSR  PLAATS
         INY
         LDA  LOBYTE
         JSR  PLAATS
         LDY  #$1F        ; en lees deze byte in
         JSR  LEES
         LDY  OPSLAG
         STA  ($FA),Y     ; plaats in gewone RAM
         CLC
         LDA  LOBYTE      ; laat pointers naar de
         ADC  #$50        ; volgende rij wijzen
         STA  LOBYTE      ; (zelfde kolom)
         BCC  CONT
         INC  HIBYTE
CONT     INY
         CPY  #$C8        ; alle 200 bytes gehad?
         BNE  LOOP
         LDX  $FC         ; indien alle bits in 1 byte
         BNE  CONT3       ; zaten : dan klaar
         LDX  #$07
         BPL  EINDE       ; gelijkwaardig met JMP
CONT3    DEX              ; bitteller 1 verlagen.
         BNE  VERDER      ; tweede nul opvangen
         LDY  #$00
LOOP1    LDA  ($FA),Y     ; alle bits 1 plaats naar
         LSR              ; rechts schuiven
         STA  ($FA),Y
         INY
         CPY  #$C8
         BNE  LOOP1
         DEC  LOBYTEOPSLAG; op volgende kolom pointen
         JMP  EINDE       ; en we zijn klaar.
VERDER   DEC  LOBYTEOPSLAG; zet pointers op volg.
         LDA  LOBYTEOPSLAG; kolom
         STA  LOBYTE
         LDA  HIBYTEOPSLAG
         STA  HIBYTE
LOOP2    LDY  #$00        ; bewaar Y
         STY  OPSLAG
         LDY  #$12        ; lees byte in
         LDA  HIBYTE
         ISR  PLAATS
         INY
         LDA  LOBYTE
         JSR  PLAATS
         LDY  #$1F
         JSR  LEES
         STA  BEWAAR      ; bewaar die byte
         LDY  OPSLAG      ; lees Y (byteteller)
         LSR  BEWAAR      ; schuif de bits bij
         LDA  ($FA),Y     ; de byte van de
         ROR              ; vorige  kolom
         STA  ($FA),Y
         DEX
         BNE  LOOP1
         LSR              ;nogmaals opschuiven
         STA  ($FA),Y     ;zodat bit 7 leeg is.
         CLC
         LDA  LOBYTE      ; pointers op volgende
         ADC  #$50        ; byte laten wijzen
         STA  LOBYTE
         BCC  CONT2
         INC  HIBYTE
CONT2    LDX  $FC         ; bitteller verlagen
         DEX
         INY              ; volgende byte
         CPY  #$C8        ; alle bytes gehad?
         BNE  LOOP2
EINDE    STX  $FC         ; bitteller opslaan
         RTS
```

listing 2

```
DRUK AF      LDY  #$00
LOOP         LDA  ($FA),Y      ; lees een byte
        ORA  #$80 ; plus $80
        JSR  $FFD2             ; druk af
        INY       ; volgende byte
        CPY  #$C8; alle 200 bytes gehad?
             BNE  LOOP         ; nee : doe verder
             RTS
```

zullen we eerst met een LSR alle bits een plaats naar rechts moeten opschuiven.

Na al deze uitleg kunnen we nu eindelijk overgaan tot het eigenlijke programma. Wel nog dit: de benamingen 'lijn' en 'kolom' zullen steeds slaan op de printeruitvoer, en niet op het beeldscherm.

```
listing 3
MAIN  LDA   #$5C             ; 92 maal de routine
      STA   COUNT            ; doorlopen
      LDA   #$4F             ; pointer op laatste kolom
      STA   LOBYTEOPSLAG     ; van de eerste rij
      LDA   #$00
      STA   HIBYTEOPSLAG
      LDA   #$00             ; bitteller op nul
      STA   $FC
LOOP  LDA   #$00             ; zet pointers op geheugen-
      STA   $FA              ; ruimte naarwaar gekopieerd
      LDA   #$04             ; moet worden.
      STA   $FB
      JSR   KOPIEER          ; kopieer deel van
      LDA   #$00             ; het scherm
      STA   $FA
      LDA   #$04
      STA   $FB
      JSR   DRUK AF          ; druk kopie af
      LDA   #$0D             ; begin nieuwe regel
      JSR   $FFD2
      DEC   COUNT            ; teller verlagen
      BNE   LOOP
      RTS
```

Laten we beginnen met de kopieerroutine (meteen ook de langste en moeilijkste routine). Deze kopieert 200 bytes (zie listing 1).

De volgende stap is het afdrukken van de gekopieerde Bytes. Dit gaat nu vrij eenvoudig, we mogen enkel niet vergeten bij alle Bytes $80 op te tellen. Deze optelling kunnen we echter ook vervangen door een logische OR met $80 (probeer dit maar)(zie listing 2).

Ons programma is nu eigenlijk af: we hebben enkel nog een routine nodig die de twee bovenstaande deelprogramma's aan elkaar knoopt.

Indien we dit programma zouden runnen, dan zouden we zien dat de afdruk redelijk veel uitgerekt is in de lengterichting. Dit kunnen we opvangen door iedere vertikale lijn twee keer af te drukken. Onze DRUK AF-routine ziet er dan als volgt uit:

```
DRUK AF   LDY   #$00
LOOP      LDA   ($FA),Y
          ORA   #$80
          JSR   $FFD2 ;druk af
          JSR   $FFD2 ;opnieuw
          INY
          CPY   #$C8
          BNE   LOOP
          RTS
```

Het nadeel van deze manier van afdrukken is evenwel dat we een minder fijne tekening krijgen (zie figuren 1 en 2). Een andere oplossing zou zijn om in de horizontale richting enkele kolommen weg te laten, en in plaats van een 640x200 tekening slechts een 320x200 tekening af te drukken. Op deze methode zullen we hier evenwel niet ingaan.

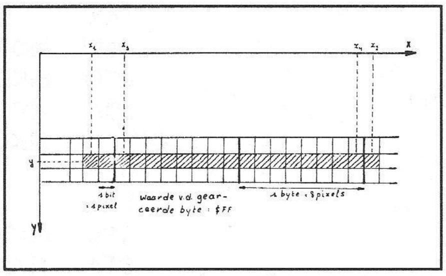
Horizontale lijnen

Indien u de demo uit het vorig nummer geprobeerd heeft, zult u ongetwijfeld gemerkt hebben dat het betrekkelijk veel tijd in beslag nam om de X- en de Y-as te trekken. Daarom zullen we hier een machinetaalroutine voor ontwerpen, zodat we in het vervolg dergelijke assen kunnen trekken in een fractie van een seconde.
Laten we beginnen met een routine voor horizontale lijnen. Op het eerste gezicht zouden we deze routine opstellen aan de hand van volgend algoritme:

```
HLINE x1 TO x2, y        met x1 < x2

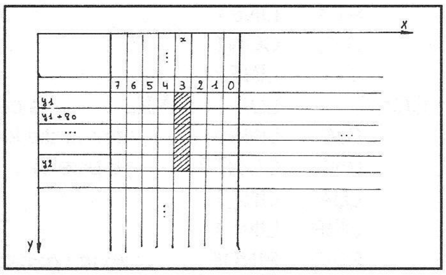
        teller := x1    (teller beslaat 2 Bytes)
LOOP PLOT (teller AND 255, teller/256,
             y)
        teller := teller + 1
        als teller <= x2 dan LOOP
        einde
```

Wat dus analoog is aan de BASIC-routine uit het eerste artikel. Het kan echter allemaal nog veel sneller: bekijken we immers eens het volgende schema:



schema 4

Indien we het lijnstuk [x3,x4] beschouwen, zien we dat dit juist overeenstemt met 2 Bytes. In plaats van 16 keer de PLOT-routine aan te roepen om een pixel aan te schakelen, zou het dus veel sneller zijn om de twee betreffende Bytes met de waarde $FF te vullen.
Wat nu gedaan met lijnstukken die niet precies overeenkomen met een geheel aantal Bytes? Beschouwen we hiertoe het lijnstuk [x1,x2], en verdelen we dit in de volgende stukken: [x1,x3] + [x3,x4] + [x4,x2]. Voor [x3,x4] kunnen we de zonet besproken methode toepassen, maar voor [x1,x3] en [x4,x2] zijn we genoodzaakt van ons eerste algoritme te gebruiken.
Het voordeel van deze werkwijze is vooral merkbaar indien men verschillende (lange) rechten moet trekken (bijvoorbeeld in een FILL-routine). Zie listing 4.



schema 5

Vertikale lijnen

Voor vertikale assen zouden we een analoge methode kunnen toepassen, zoals schema 5 duidelijk maakt.

Het enige wat we hier moeten doen is berekenen welke bit geset dient te worden (in ons voorbeeld is dit de derde), en deze setten in iedere rij. In ons voorbeeld dienen we dus in iedere rij de derde bit te setten, en dit kan door met iedere rij een OR uit te voeren met %00001000 = $08.
Dit alles zal echter heel wat programmeerwerk vergen, en men kan zich afvragen of het wel de moeite waard is het VLINE-programma te versnellen als men al een versnelde HILNE heeft die kan instaan voor zaken zoals bijvoorbeeld FILL. We zullen dan ook niet nader ingaan op bovenstaande methode, maar een veel eenvoudiger programma in de plaats geven. Zo eenvoudig zelfs dat verdere uitleg overbodig is.

Listing 4

aanroeping:

HLINE X1(LO), X1(HI), Y
X2(LO) --> LINE4
X2(HI) --> LINE5

```
HLINE       STA     LINE1
      STX   LINE2
      STY   LINE3

      ; eerste deel: trek [x1,x3]
      AND   #$07        ; bereken 7 - (X and 7)
      BEQ   KLAAR       ; indien nul : klaar
      STA   TIJDELIJK
      LDA   #$07
      SEC
      SBC   TIJDELIJK
      STA   TIJDELIJK
      LDA   LINE1
      LDX   LINE2
      LDY   LINE3
      JSR   PLOTIN      ; druk punt
      LDA   LINE1
      ADC   #$01        ; x-coordinaat verhogen
      STA   LINE1
      BCC   CONT
      INC   LINE2
CONT        LDA     LINE2       ; is de lijn al klaar?
      CMP   LINE5       ; (dan is die kleiner dan
      BNE   CONT2       ; 8 pixels)
      LDA   LINE1
      CMP   LINE4
      BEQ   EINDE       ; zo ja : gedaan
CONT2       DEC     TIJDELIJK   ; anders : volgende dot
      BPL   LOOP

      ; tweede deel : trek [x3,x4]
      ; bepaal aantal af te drukken bytes
      ; formule:
      ; N = INT ( (x2-x3+1)/8 )

KLAAR       SEC             ; bereken x2-x3+1
      LDA LINE4
      SBC LINE1
      STA TIJDELIJK
      LDA LINE5
      SBC LINE
      TAX
      LDA TIJDELIJK
      ADC #$01
      STA TIJDELIJK
      BCC CONT
      INX
      TXA
      LSR           ; en deel door 8
      ROR TIJDELIJK
      LSR
      ROR TIJDELIJK
      LSR
      ROR   TIJDELIJK
      DEC   TIJDELIJK
      BEQ   VERDER

      ; in de accu zit nu het aantal keer dat een
      ; volledige byte mag geset worden.
      ; indien dit nul maal is, wordt er naar
      ; deel 3 gesprongen.

      ; set de bytes
      LDA   LINE1       ; we berekenen adres
      STA   EERSTE      ; van byte die moet
      LDX   LINE2       ; behandeld worden.
      STX   TWEEDE
      LDY   LINE3
      STY   DERDE
      JSR   BEREKEN BYTE    ; (resultaat in derde
                            ; en vierde)
      LDY   #$12
      LDA   VIERDE
      JSR   PLAATS
      INY
      LDA   DERDE
      JSR   PLAATS
      LDA   #$FF        ; vul Byte met $FF
      LDY   #$1F
      JSR   PLAATS
      CLC
      LDA   DERDE       ; zet pointers op
      ADC   #$01        ; volgende Byte
      STA   DERDE
      BCC   CONT
      INC   VIERDE
CONT        DEC     TIJDELIJK
      BPL   LOOP

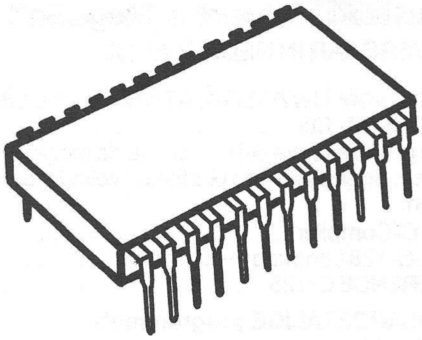
      ; derde deel : trek [x4,x2]

VERDER      LDA     LINE4       ;bereken X2 and 7
      AND   #$07        ;als dit 7 is, dan moet
      TAX               ;volledige byte geset
      CMP   #$07        ;worden, dit is gedaan
      BEQ   EINDE       ;in deel 2 => EINDE
      LDA   #$00
LOOP        SEC             ;plaats in accu zoveel
      ROR               ;enen als dots geset
      DEX               ;moeten worden
      BPL   LOOP
      STA   TIJDELIJK
      LDA   LINE4       ;en plaats de accu in
      STA   EERSTE      ;desbetreffende byte
      LDA   LINE5
      STA   TWEEDE
      LDY   LINE3
      STY   DERDE
      JSR   BEREKEN BYTE
      JSR   $0B6A
      ORA   $28
      JSR   $0BC6
EINDE       RTS
```

Listing 5

Aanroeping:

VLINE X(lo), X(hi), Y1
Y2 ---> LINE4



```
VLINE STA   LINE1
      STX   LINE2
      STY   LINE3
      LDA   LINE4
      SEC
      SBC   LINE3        ; bereken y2 - y1 + 1
      STA   TIJDELIJK    ; en plaats dit in
      INC   TIJDELIJK    ; tijdelijk.
LOOP  LDA   LINE1
      LDX   LINE2
      LDY   LINE3
      JSR   PLOT         ; druk punt af
      INC   LINE3        ; y-coörd. verhogen
      DEC   TIJDELIJK
      BNE   LOOP
      RTS
```

Kleur

Tot nu toe waren al onze tekeningen in zwart-wit. Het is natuurlijk ook mogelijk om met andere kleuren te werken. Hiervoor dienen we register 26 van de VDC-chip aan te spreken, en wel als volgt:

bits 0 - 3 : achtergrondkleur
bits 4 - 7 : voorgrondkleur

Indien v het nummer van de voorgrondkleur is, en a dit van de achtergrondkleur voorstelt, dan dienen we het volgende getal in register 26 te plaatsen:

getal = v*16 + a.

De routine in listing 6 doet dit.

De kleurnummers zijn jammer genoeg niet dezelfde als de gewone 80-koloms-kleurnummers. Ze zijn:

1 : zwart
2 : donker blauw
3 : licht blauw
4 : donker groen
5 : licht groen
6 : donker cyaan
7 : licht cyaan
8 : donker rood
9 : licht rood
10 : donker paars
11 : licht paars
12 : bruin
13 : geel
14 : grijs
15 : wit

Het programma

Het programma uit het vorig nummer had de vervelende eigenschap dat bij veel computers aan de rechterzijde van het scherm strepen merkbaar waren. Dit komt omdat er twee versies van de CBM 128 in omloop zijn. Bij de ene versie moeten de bits 0 tot 2 van register $19 gecleard zijn bij het aanschakelen van de hoge resolutie, en bij de andere moeten deze geset zijn. Indien de strepen bij u merkbaar waren, bezit u de tweede versie, en dient u register $19 met $87 te laden in plaats van met $80.

Het PLOT-programma uit het vorig nummer is enigszins gewijzigd, zodat er nu ook een UNPLOT voorhanden is. Daarom wordt het hele programma nog eens afgedrukt, samen met het programma van deze maand.

De verschillende routines dienen aangeroepen te worden zoals in listing 7 is vermeld.

Tenslotte voor de duidelijkheid een overzicht van de programma's die hierna volgen: listing 1 is de BASIC-lader, listings 2 en 3 zijn demo-programma's, en indien u het programma liever met de monitor ingeeft, dan dient u listing 4 in plaats van listing 1 in te typen.

Peter Heinckiens

Listing 6

Aanroeping:

COLOR v,a

met v = voorgrondkleur
a = achtergrondkleur.

```
COLOR ASL            ; bereken v*16
      ASL
      ASL
      ASL
      STX   BEWAAR   ; a in bewaar
      ORA   BEWAAR   ; tel a bij accu op
      LDY   #$26     ; plaats accu in
      JSR   PLAATS   ; register 26
      RTS
```

Listing 7

```
AAN      :  SYS DEC ("B18")
UIT      :  SYS DEC ("B3A")
WIS      :  SYS DEC ("B22")
PLOT     :  SYS DEC ("BB3"), X(LO), X(HI), Y
UNPLOT   :  SYS DEC ("BBB"), X(LO), X(HI), Y
HARDCOPY :  SYS DEC ("C98")
HLINE    :  POKE DEC ("1153"), X2(LO)
            POKE DEC ("1154"), X2(HI)
            SYS DEC ("CCD"), X1(LO), X1(HI), Y
VLINE    :  POKE DEC ("1153"), Y2
            SYS DEC ("D9B"), X(LO), X(HI), Y1
COLOR    :  SYS DEC ("DC4"), V, A
```

Programmalistings minigraph

```
10 REM  ==============================================================
11 REM
12 REM           MINIGRAPH  -  DEEL 1 + 2     (C) 1986
13 REM
14 REM                     PETER HEINCKIENS
15 REM                     BORMTESTRAAT 20
16 REM                     9100 LOKEREN  (BELGIE)
17 REM                     091/48.03.85
18 REM
19 REM  ==============================================================
20 FOR T=0 TO 724
21 READ A
22 POKE 2816 + T, A
23 B = B + A
24 NEXT
25 IF B<>80387 THEN PRINT "  FOUT IN DATA !"
26 DATA 140 , 0 , 214 , 44 , 0 , 214 , 16 , 251 , 141 , 1
27 DATA 214 , 96 , 140 , 0 , 214 , 44 , 0 , 214 , 16 , 251
28 DATA 173 , 1 , 214 , 96 , 160 , 25 , 169 , 135 , 32 , 0
29 DATA 11 , 32 , 34 , 11 , 162 , 64 , 160 , 18 , 138 , 32
30 DATA 0 , 11 , 160 , 31 , 169 , 0 , 32 , 0 , 11 , 160
31 DATA 30 , 32 , 0 , 11 , 202 , 16 , 235 , 96 , 160 , 25
32 DATA 169 , 64 , 32 , 0 , 11 , 32 , 12 , 206 , 162 , 59
33 DATA 76 , 220 , 225 , 133 , 250 , 134 , 251 , 132 , 252 , 72
34 DATA 32 , 125 , 11 , 104 , 56 , 41 , 7 , 133 , 250 , 169
35 DATA 7 , 229 , 250 , 168 , 169 , 1 , 192 , 0 , 240 , 4
36 DATA 10 , 136 , 208 , 252 , 133 , 250 , 160 , 18 , 165 , 253
37 DATA 32 , 0 , 11 , 200 , 165 , 252 , 32 , 0 , 11 , 160
38 DATA 31 , 32 , 12 , 11 , 96 , 70 , 251 , 102 , 250 , 70
39 DATA 251 , 102 , 250 , 70 , 251 , 102 , 250 , 24 , 169 , 0
40 DATA 133 , 253 , 165 , 252 , 160 , 4 , 101 , 252 , 144 , 3
41 DATA 230 , 253 , 24 , 136 , 208 , 246 , 10 , 38 , 253 , 10
42 DATA 38 , 253 , 10 , 38 , 253 , 10 , 38 , 253 , 101 , 250
43 DATA 133 , 252 , 165 , 251 , 101 , 253 , 133 , 253 , 96 , 32
44 DATA 73 , 11 , 5 , 250 , 76 , 198 , 11 , 32 , 73 , 11
45 DATA 133 , 251 , 165 , 250 , 73 , 255 , 37 , 251 , 72 , 160
46 DATA 18 , 165 , 253 , 32 , 0 , 11 , 200 , 165 , 252 , 32
47 DATA 0 , 11 , 104 , 160 , 31 , 32 , 0 , 11 , 96 , 173
48 DATA 80 , 17 , 141 , 82 , 17 , 173 , 81 , 17 , 141 , 83
49 DATA 17 , 160 , 0 , 132 , 254 , 160 , 18 , 173 , 83 , 17
50 DATA 32 , 0 , 11 , 200 , 173 , 82 , 17 , 32 , 0 , 11
51 DATA 160 , 31 , 32 , 12 , 11 , 164 , 254 , 145 , 250 , 24
52 DATA 173 , 82 , 17 , 105 , 80 , 141 , 82 , 17 , 144 , 3
53 DATA 238 , 83 , 17 , 200 , 192 , 200 , 208 , 211 , 166 , 252
54 DATA 208 , 4 , 162 , 7 , 16 , 99 , 202 , 208 , 18 , 160
55 DATA 0 , 177 , 250 , 74 , 145 , 250 , 200 , 192 , 200 , 208
56 DATA 246 , 206 , 80 , 17 , 76 , 129 , 12 , 206 , 80 , 17
57 DATA 173 , 80 , 17 , 141 , 82 , 17 , 173 , 81 , 17 , 141
58 DATA 83 , 17 , 160 , 0 , 132 , 254 , 160 , 18 , 173 , 83
59 DATA 17 , 32 , 0 , 11 , 200 , 173 , 82 , 17 , 32 , 0
60 DATA 11 , 160 , 31 , 32 , 12 , 11 , 133 , 253 , 164 , 254
61 DATA 70 , 253 , 177 , 250 , 106 , 145 , 250 , 202 , 208 , 246
62 DATA 74 , 145 , 250 , 24 , 173 , 82 , 17 , 105 , 80 , 141
63 DATA 82 , 17 , 144 , 3 , 238 , 83 , 17 , 166 , 252 , 202
64 DATA 200 , 192 , 200 , 208 , 195 , 134 , 252 , 96 , 234 , 160
65 DATA 0 , 177 , 250 , 9 , 128 , 32 , 210 , 255 , 32 , 210
66 DATA 255 , 200 , 192 , 200 , 208 , 241 , 96 , 234 , 169 , 92
67 DATA 141 , 84 , 17 , 169 , 79 , 141 , 80 , 17 , 169 , 0
68 DATA 141 , 81 , 17 , 169 , 0 , 133 , 252 , 169 , 0 , 133
69 DATA 250 , 169 , 4 , 133 , 251 , 32 , 219 , 11 , 169 , 0
70 DATA 133 , 250 , 169 , 4 , 133 , 251 , 32 , 133 , 12 , 169
71 DATA 13 , 32 , 210 , 255 , 206 , 84 , 17 , 208 , 224 , 96
72 DATA 234 , 141 , 80 , 17 , 142 , 81 , 17 , 140 , 82 , 17
73 DATA 41 , 7 , 240 , 54 , 133 , 40 , 169 , 7 , 56 , 229
74 DATA 40 , 133 , 40 , 173 , 80 , 17 , 174 , 81 , 17 , 172
75 DATA 82 , 17 , 32 , 179 , 11 , 173 , 80 , 17 , 105 , 1
76 DATA 141 , 80 , 17 , 144 , 3 , 238 , 81 , 17 , 173 , 81
77 DATA 17 , 205 , 84 , 17 , 208 , 8 , 173 , 80 , 17 , 205
78 DATA 83 , 17 , 240 , 191 , 198 , 40 , 16 , 211 , 56 , 173
79 DATA 83 , 17 , 237 , 80 , 17 , 133 , 40 , 173 , 84 , 17
80 DATA 237 , 81 , 17 , 170 , 165 , 40 , 105 , 1 , 133 , 40
81 DATA 144 , 1 , 232 , 138 , 74 , 102 , 40 , 74 , 102 , 40
82 DATA 74 , 102 , 40 , 198 , 40 , 240 , 53 , 173 , 80 , 17
83 DATA 133 , 250 , 174 , 81 , 17 , 134 , 251 , 172 , 82 , 17
84 DATA 132 , 252 , 32 , 125 , 11 , 160 , 18 , 165 , 253 , 32
85 DATA 0 , 11 , 200 , 165 , 252 , 32 , 0 , 11 , 169 , 255
86 DATA 160 , 31 , 32 , 0 , 11 , 24 , 165 , 252 , 105 , 1
```

Programmalistings minigraph

```
87 DATA 133 , 252 , 144 , 2 , 230 , 253 , 198 , 40 , 16 , 221
88 DATA 173 , 83 , 17 , 41 , 7 , 170 , 201 , 7 , 240 , 35
89 DATA 169 , 0 , 56 , 106 , 202 , 16 , 251 , 133 , 40 , 173
90 DATA 83 , 17 , 133 , 250 , 173 , 84 , 17 , 133 , 251 , 172
91 DATA 82 , 17 , 132 , 252 , 32 , 125 , 11 , 32 , 106 , 11
92 DATA 5 , 40 , 32 , 198 , 11 , 96 , 234 , 141 , 80 , 17
93 DATA 142 , 81 , 17 , 140 , 82 , 17 , 173 , 83 , 17 , 56
94 DATA 237 , 82 , 17 , 133 , 40 , 230 , 40 , 173 , 80 , 17
95 DATA 174 , 81 , 17 , 172 , 82 , 17 , 32 , 179 , 11 , 238
96 DATA 82 , 17 , 198 , 40 , 208 , 237 , 96 , 234 , 10 , 10
97 DATA 10 , 10 , 134 , 251 , 5 , 251 , 160 , 26 , 32 , 0
98 DATA 11 , 96 , 234 , 234 , 0


READY.
```

LISTING 1

```
 10 REM ====================================================
 11 REM
 12 REM              MINIGRAPH    -     DEMO 1
 13 REM
 14 REM                       DOOR
 15 REM
 16 REM                 PETER HEINCKIENS
 17 REM
 18 REM                     (C) 1986
 19 REM
 20 REM ====================================================
 21 :
 22 FAST
 23 :
 24 REM   PARAMETERS
 25 A% = 255 : B% = 256
 26 X% = 62  : Y% = 100
 27 P% = DEC ("BB3")
 28 YA% = DEC("0D9B")
 29 XA% = DEC("0CCD")
 30 DEFFNF(X)=SIN(X)-COS(2*X)
 31 :
 32 REM   BITMAP AAN
 33 SYS DEC("0B18")
 34 :
 35 REM   TEKEN X-AS
 36 POKE DEC ("1153"), 639 AND A%
 37 POKE DEC ("1154"), 639 / B%
 38 SYS XA%, 0, 0, 100
 39 :
 40 REM   TEKEN Y-AS
 41 POKE DEC("1153"),199
 42 SYS YA%, 64, 1, 0
 43 :
 44 REM   VERTIKALE VERDELING
 45 FOR T=0 TO 639 STEP 40
 46 POKE DEC("1153"), 104
 47 SYS YA%, T AND A%, T/B% ,96
 48 NEXT
 49 :
 50 POKE DEC("1153"), 104
 51 SYS YA%, 639 AND A%, 639/B% ,96
 52 :
 53 REM   HORIZONTALE VERDELING
 54 FOR T=0 TO 199 STEP 20
 55 POKE DEC("1153"),67
 56 POKE DEC("1154"),1
 57 SYS XA%, 59, 1, T
 58 NEXT
 59 :
 60 POKE DEC("1153"),67
 61 POKE DEC("1154"),1
 62 SYS XA%, 59, 1, 199
 63 :
 64 REM   TEKEN FUNCTIE
 65 FOR T = 0 TO 639
 66 :  X = 0.01*(T-320)
 67 :  SYS P%, T AND A%, T/B%, 100-FNF(X)*30
 68 NEXT
 69 :
 70 REM   HARDCOPY
 71 GETKEY A$ : IF A$ <> "P" THEN 76
 72 OPEN 4,4 : CMD4 : PRINT CHR$(8)
 73 SYS DEC("C98")
 74 PRINT#4 : CLOSE4
 75 :
 76 REM   BITMAP UIT
 77 GETKEYA$
 78 SYS DEC ("0B3A")
 79 PRINT"□"

READY.
```

LISTING 2

Het restant volgt in het volgende nummer.